

#2

Docket No. 826.1705

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

J1017 U.S. PTO
09/812562
03/21/01

In re Patent Application of:)
Tadahiro UEHARA, et al.)
Serial No.: To be assigned) Group Art Unit: Unassigned
Filed: March 16, 2001) Examiner: Unassigned
For: APPARATUS FOR SWITCHING)
AN OBJECT MANAGING)
METHOD AND A METHOD)
THEREOF)

**SUBMISSION OF CERTIFIED COPY OF PRIOR FOREIGN
APPLICATION IN ACCORDANCE
WITH THE REQUIREMENTS OF 37 C.F.R. §1.55**

*Assistant Commissioner for Patents
Washington, D.C. 20231*

Sir:

In accordance with the provisions of 37 C.F.R. §1.55, the applicant submits herewith a certified copy of the following foreign application:

Japanese Patent Application No. 2000-129655
Filed: April 28, 2000.

It is respectfully requested that the applicant be given the benefit of the foreign filing date as evidenced by the certified papers attached hereto, in accordance with the requirements of 35 U.S.C. §119.

Respectfully submitted,

STAAS & HALSEY LLP

Date: March 16, 2001

By: _____

James D. Halsey, Jr.
Registration No. 22,729

700 Eleventh Street, N.W.
Suite 500
Washington, D.C. 20001
(202) 434-1500

PATENT OFFICE
JAPANESE GOVERNMENT

j1017 U.S. PTO
09/812562
03/21/01

This is to certify that the annexed is a true copy of the following application as filed with this Office.

Date of Application: April 28, 2000

Application Number: Patent Application
No. 2000-129655

Applicant(s): FUJITSU LIMITED

December 22, 2000

Commissioner,
Patent Office Kozo Oikawa

Certificate No. 2000-3105864

日本国特許庁

PATENT OFFICE
JAPANESE GOVERNMENT

J1017 U.S. PTO
09/812562



別紙添付の書類に記載されている事項は下記の出願書類に記載されて
いる事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed
with this Office.

出願年月日

Date of Application:

2000年 4月28日

出願番号

Application Number:

特願2000-129655

願人

Applicant(s):

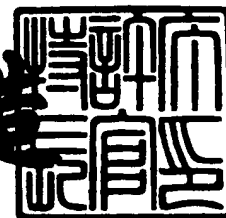
富士通株式会社

CERTIFIED COPY OF
PRIORITY DOCUMENT

2000年12月22日

特許庁長官
Commissioner,
Patent Office

及川耕造



出証番号 出証特2000-3105864

【書類名】 特許願

【整理番号】 0050891

【提出日】 平成12年 4月28日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 9/44
G06F 17/60

【発明の名称】 オブジェクト管理方法を切り替える装置および方法

【請求項の数】 15

【発明者】

 【住所又は居所】 神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内

 【氏名】 上原 忠弘

【発明者】

 【住所又は居所】 神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内

 【氏名】 山本 里枝子

【発明者】

 【住所又は居所】 神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内

 【氏名】 小田切 良弘

【発明者】

 【住所又は居所】 神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内

 【氏名】 村林 雅隆

【特許出願人】

 【識別番号】 000005223

 【氏名又は名称】 富士通株式会社

【代理人】

 【識別番号】 100074099

【住所又は居所】 東京都千代田区二番町 8 番地 2 0 二番町ビル 3 F

【弁理士】

【氏名又は名称】 大菅 義之

【電話番号】 03-3238-0031

【選任した代理人】

【識別番号】 100067987

【住所又は居所】 神奈川県横浜市鶴見区北寺尾 7 - 2 5 - 2 8 - 5 0 3

【弁理士】

【氏名又は名称】 久木元 彰

【電話番号】 045-573-3683

【手数料の表示】

【予納台帳番号】 012542

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9705047

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 オブジェクト管理方法を切り替える装置および方法

【特許請求の範囲】

【請求項 1】 1 つ以上のコンポーネントを含むアプリケーションプログラムにより用いられるオブジェクトを、該アプリケーションプログラムを構成し実行する基盤となるコンポーネント基盤の中で管理するオブジェクト管理装置であって、

あらかじめ用意された複数のオブジェクト管理方法から、コンポーネントの種類に適したオブジェクト管理方法を選択する選択手段と、

前記コンポーネント基盤の中で、外部記憶のデータに対応する永続化オブジェクトを特定のアルゴリズムでキャッシュするオブジェクトキャッシュ部品と、該外部記憶のデータと該永続化オブジェクトの間の変換を行うオブジェクト永続化処理部品のうち、少なくとも一方を、選択されたオブジェクト管理方法に対応する部品に切り替える切替手段と

を備えることを特徴とするオブジェクト管理装置。

【請求項 2】 前記切替手段は、前記選択されたオブジェクト管理方法に対応する部品のプログラムを自動生成して、前記オブジェクト管理装置に組み込むことを特徴とする請求項 1 記載のオブジェクト管理装置。

【請求項 3】 前記切替手段は、あらかじめ用意された部品群の中から、前記選択されたオブジェクト管理方法に対応する部品を選択して、前記オブジェクト管理装置に組み込むことを特徴とする請求項 1 記載のオブジェクト管理装置。

【請求項 4】 前記選択手段は、ユーザが指定したオブジェクト管理方法を選択することを特徴とする請求項 1 記載のオブジェクト管理装置。

【請求項 5】 前記選択手段は、前記アプリケーションプログラムが指定したオブジェクト管理方法を選択することを特徴とする請求項 1 記載のオブジェクト管理装置。

【請求項 6】 前記選択手段は、コンポーネントの用途情報を入力する入力手段を含み、該用途情報に基づいてオブジェクト管理方法を選択することを特徴とする請求項 1 記載のオブジェクト管理装置。

【請求項 7】 前記切替手段は、トランザクション毎にオブジェクトをキャッシュするキャッシュテーブルを有し、トランザクションの開始後に該キャッシュテーブルにオブジェクトを登録し、該トランザクションの終了時に該キャッシュテーブルをクリアするようなオブジェクトキャッシュ部品を、前記選択されたオブジェクト管理方法に対応する部品として用いることを特徴とする請求項 1 記載のオブジェクト管理装置。

【請求項 8】 前記切替手段は、全トランザクション共通のキャッシュテーブルを有し、1つのトランザクションの終了時に該キャッシュテーブルをクリアせず、他のトランザクションに該キャッシュテーブル上のオブジェクトを利用させるようなオブジェクトキャッシュ部品を、前記選択されたオブジェクト管理方法に対応する部品として用いることを特徴とする請求項 1 記載のオブジェクト管理装置。

【請求項 9】 前記切替手段は、前記アプリケーションプログラムが処理対象となるオブジェクトの取得を要求したとき、該アプリケーションプログラムが使用していたオブジェクトのデータを前記外部記憶に書込み、次のデータを該使用していたオブジェクトに設定して該アプリケーションプログラムに返すようなオブジェクトキャッシュ部品を、前記選択されたオブジェクト管理方法に対応する部品として用いることを特徴とする請求項 1 記載のオブジェクト管理装置。

【請求項 10】 前記切替手段は、前記アプリケーションプログラムが要求する前に、前記外部記憶のすべてのデータをあらかじめキャッシュしておくようなオブジェクトキャッシュ部品を、前記選択されたオブジェクト管理方法に対応する部品として用いることを特徴とする請求項 1 記載のオブジェクト管理装置。

【請求項 11】 前記切替手段は、前記永続化オブジェクトに関連するインタフェースを登録し、該インタフェースを該永続化オブジェクトとともに再利用するようなオブジェクトキャッシュ部品を、前記選択されたオブジェクト管理方法に対応する部品として用いることを特徴とする請求項 1 記載のオブジェクト管理装置。

【請求項 12】 前記コンポーネントの種類に適した 1 つ以上のデータ項目を指定する指定手段をさらに備え、前記切替手段は、前記永続化オブジェクトの

指定されたデータ項目について、前記外部記憶のデータと該永続化オブジェクトの間の変換を行うようなオブジェクト永続化処理部品を、前記選択されたオブジェクト管理方法に対応する部品として用いることを特徴とする請求項 1 記載のオブジェクト管理装置。

【請求項 1 3】 1 つ以上のコンポーネントを含むアプリケーションプログラムにより用いられるオブジェクトを、該アプリケーションプログラムを構成し実行する基盤となるコンポーネント基盤の中で管理するオブジェクト管理装置であって、

あらかじめ用意された複数のオブジェクト管理方法から、コンポーネントの種類に適したオブジェクト管理方法を選択するためのスケジュール情報を格納する格納手段と、

前記コンポーネント基盤の中で、外部記憶のデータに対応する永続化オブジェクトを特定のアルゴリズムでキャッシュするオブジェクトキャッシュ部品と、該外部記憶のデータと該永続化オブジェクトの間の変換を行うオブジェクト永続化処理部品のうち、少なくとも一方を、前記スケジュール情報に従って、前記適したオブジェクト管理方法に対応する部品に切り替える切替手段と

を備えることを特徴とするオブジェクト管理装置。

【請求項 1 4】 1 つ以上のコンポーネントを含むアプリケーションプログラムにより用いられるオブジェクトを、該アプリケーションプログラムを構成し実行する基盤となるコンポーネント基盤の中で管理するコンピュータのためのプログラムを記録した記録媒体であって、

記録されたプログラムは、

あらかじめ用意された複数のオブジェクト管理方法から、コンポーネントの種類に適したオブジェクト管理方法を選択し、

前記コンポーネント基盤の中で、外部記憶のデータに対応する永続化オブジェクトを特定のアルゴリズムでキャッシュするオブジェクトキャッシュ部品と、該外部記憶のデータと該永続化オブジェクトの間の変換を行うオブジェクト永続化処理部品のうち、少なくとも一方を、選択されたオブジェクト管理方法に対応する部品に切り替える

処理を前記コンピュータに実行させることを特徴とするコンピュータ読み取り可能な記録媒体。

【請求項 15】 1つ以上のコンポーネントを含むアプリケーションプログラムにより用いられるオブジェクトを、該アプリケーションプログラムを構成し実行する基盤となるコンポーネント基盤の中で管理する管理方法であって、

複数のオブジェクト管理方法をあらかじめ用意し、

前記複数のオブジェクト管理方法からコンポーネントの種類に適したオブジェクト管理方法を選択し、

前記コンポーネント基盤の中で、外部記憶のデータに対応する永続化オブジェクトを特定のアルゴリズムでキャッシュするオブジェクトキャッシュ部品と、該外部記憶のデータと該永続化オブジェクトの間の変換を行うオブジェクト永続化処理部品のうち、少なくとも一方を、選択されたオブジェクト管理方法に対応する部品に、自動的に切り替え、

前記選択されたオブジェクト管理方法に対応する部品を用いて、前記アプリケーションプログラムにより用いられるオブジェクトを管理する

ことを特徴とする管理方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、オブジェクト指向プログラミングに係り、コンポーネントを組み合わせさせてアプリケーションプログラムを構築し運用するためのオブジェクト管理装置およびその方法に関する。

【0002】

【従来の技術】

業務システム等のアプリケーションプログラム（アプリケーション）は、しばしば、コンポーネントと呼ばれるプログラム単位を組み合わせられて構築される。このようなアプリケーションの構築時／運用時（実行時）において、コンポーネント基盤は、二次記憶アクセス機能（例えば、データベースアクセス機能）、トラザクション機能、分散環境における通信機能等を各コンポーネントに提供する

【 0 0 0 3 】

このうち、二次記憶アクセス機能は、アプリケーションからの要求に従って二次記憶にアクセスし、コンポーネントが指定する永続化オブジェクトのデータの書込みや読出しを行う機能である。ここで、永続化オブジェクトとは、二次記憶内のデータに対応するオブジェクトを指す。このオブジェクトは、通常、二次記憶内のデータと同じくらい長い寿命を持つ。

【 0 0 0 4 】

二次記憶への書込み・読出しの頻度はアプリケーション全体の性能に大きく関わるため、二次記憶へのアクセス回数を削減するために、オブジェクトのキャッシング機構がよく用いられる。この機構では、過去に二次記憶から読込んだデータを持つオブジェクトをメモリ上のキャッシュ内に蓄積しておき、同じデータの検索要求があった場合には、キャッシュ内のオブジェクトが用いられる。

【 0 0 0 5 】

【発明が解決しようとする課題】

しかしながら、上述した従来のコンポーネント基盤におけるキャッシングには、次のような問題がある。

【 0 0 0 6 】

オブジェクトのキャッシングでは、キャッシュ内のデータの有効期間が重要になる。例えば、オブジェクトがキャッシュに登録されたあと、そのキャッシュを利用しないアプリケーションが、登録されたオブジェクトに対応する二次記憶のデータを変更すると、キャッシュ内のオブジェクトのデータと二次記憶内のデータが一致しなくなる。

【 0 0 0 7 】

そこで、この場合は、1つのトランザクション内に限定したキャッシングを行い、トランザクション終了時にデータが変更されている場合は、そのトランザクションの結果を無効にするような機構が必要となる。

【 0 0 0 8 】

一方、二次記憶内のデータがまったく変更されない場合には、オブジェクトの

データと二次記憶内のデータが常に等しいことが保証されるため、キャッシュに登録されたオブジェクトは常に有効となる。

【 0 0 0 9 】

従来のコンポーネント基盤が有するキャッシング機構は、すべてのアプリケーションが安全に動作できるような、効果が少ないキャッシング機構と、コンポーネント基盤自体に課された利用制限の下で最適化されたキャッシング機構のうちのいずれかである。このため、個々のアプリケーションに適したキャッシング戦略を実現するには、アプリケーション側でキャッシング機構を持つしかなく、アプリケーション開発の負荷が増大する。

【 0 0 1 0 】

本発明の課題は、コンポーネントを組み合わせてアプリケーションを構築し運用するシステムにおいて、個々のアプリケーションに適したキャッシング機構をコンポーネント基盤の中で提供するオブジェクト管理装置およびその方法を提供することである。

【 0 0 1 1 】

【課題を解決するための手段】

図 1 は、本発明のオブジェクト管理装置の原理図である。図 1 のオブジェクト管理装置は、選択手段 1 と切替手段 2 を備え、1 つ以上のコンポーネントを含むアプリケーションプログラムにより用いられるオブジェクトを、アプリケーションプログラムを構成し実行する基盤となるコンポーネント基盤の中で管理する。

【 0 0 1 2 】

選択手段 1 は、あらかじめ用意された複数のオブジェクト管理方法から、コンポーネントの種類に適したオブジェクト管理方法を選択する。切替手段 2 は、コンポーネント基盤の中で、外部記憶のデータに対応する永続化オブジェクトを特定のアルゴリズムでキャッシュするオブジェクトキャッシュ部品と、外部記憶のデータと永続化オブジェクトの間の変換を行うオブジェクト永続化処理部品のうち、少なくとも一方を、選択されたオブジェクト管理方法に対応する部品に切り替える。

【 0 0 1 3 】

オブジェクト管理方法は、システム内で使用されるオブジェクトを管理するアルゴリズムを表し、外部記憶に格納されたデータへのアクセス方法を含む。外部記憶は、例えば、二次記憶に対応する。

【0014】

選択手段1は、選択したオブジェクト管理方法の情報を、直接、あるいは格納部を介して、切替手段2に通知する。切替手段2は、通知されたオブジェクト管理方法に従って、コンポーネント基盤が提供するオブジェクトキャッシュ部品とオブジェクト永続化処理部品の少なくとも一方を、コンポーネントの種類に適した部品に切り替える。オブジェクト管理装置は、こうして切り替えられた部品を用いて、永続化オブジェクトをキャッシュする。

【0015】

このようなオブジェクト管理装置によれば、コンポーネント基盤の中で、コンポーネントの種類毎に、適切なオブジェクトキャッシュ部品およびオブジェクト永続化処理部品が自動的に組み込まれるので、個々のアプリケーションに適したキャッシング機構が自動的に提供される。

【0016】

例えば、図1の選択手段1と切替手段2は、それぞれ、後述する図2の選択部36と切替部37に対応する。

【0017】

【発明の実施の形態】

以下、図面を参照しながら、本発明の実施の形態を詳細に説明する。

本実施形態においては、コンポーネント基盤が提供するキャッシング機構等のオブジェクト管理方法を、自動的にアプリケーションの性質に適したものに切り替える。アプリケーションの性質としては、そのアプリケーションが処理するデータの用途や処理方法等が含まれる。オブジェクト管理方法をアプリケーションに合わせて切り替えることで、これまでアプリケーション側で用意していたキャッシング機構が不要になり、アプリケーション開発の工数が削減される。

【0018】

図2は、クライアントーサーバ間で情報処理を行う実施形態のシステムの基本

構成図である。図 2 のシステムは、アプリケーション 1 1、コンポーネント基盤 1 2、および二次記憶 1 3 を備える。アプリケーション 1 1 は、クライアントアプリケーション 2 1 とコンポーネント 2 2 を含み、コンポーネント 2 2 は、インタフェース 2 3 および永続化オブジェクト 2 4 を含む。

【 0 0 1 9 】

クライアントアプリケーション 2 1 は、コンポーネント 2 2 のインタフェース 2 3 を介して、コンポーネント基盤 1 2 にオブジェクトの検索、格納等の処理を依頼する。

【 0 0 2 0 】

また、コンポーネント基盤 1 2 は、例えば、サーバのメモリ上に構築され、オブジェクト管理部 3 1 と二次記憶インタフェース 3 2 を含む。オブジェクト管理部 3 1 は、インタフェース 3 3、オブジェクトキャッシュ部品 3 4、オブジェクト永続化処理部品 3 5、選択部 3 6、および切替部 3 7 を含む。

【 0 0 2 1 】

コンポーネント基盤 1 2 は、アプリケーション 1 1 を構築し実行するための基盤となり、オブジェクト管理部 3 1 は、アプリケーション 1 1 からの要求に応じて、二次記憶 1 3 内のデータに対応する永続化オブジェクト 2 4 を検索したり、格納したりする処理を行う。また、二次記憶インタフェース 3 2 は、オブジェクト管理部 3 1 と二次記憶 1 3 の間の入出力インタフェースとして動作する。二次記憶 1 3 には、例えば、レコードの形式でデータが格納される。

【 0 0 2 2 】

インタフェース 3 3 は、アプリケーション 1 1 が永続化オブジェクト 2 4 を取得・格納する方法を規定し、インタフェース 2 3 とオブジェクトキャッシュ部品 3 4 の間のインタフェースとして動作する。オブジェクトキャッシュ部品 3 4 は、性能向上のために、オブジェクトの種類毎に特定の戦略でオブジェクトをキャッシュすることで、二次記憶 1 3 へのアクセスの回数を削減する。オブジェクト永続化処理部品 3 5 は、二次記憶 1 3 内のデータとオブジェクトの間の変換を行う。

【 0 0 2 3 】

選択部 3 6 は、例えば、ユーザからの指示に基づいて、あらかじめ用意された複数のオブジェクト管理方法の中から、コンポーネント 2 2 の種類に適した方法を選択する。切替部 3 7 は、選択されたオブジェクト管理方法に対応するオブジェクトキャッシュ部品 3 4 とオブジェクト永続化処理部品 3 5 を配置して、コンポーネント 2 2 に適したキャッシング機構を自動的に生成する。これにより、オブジェクト管理部 3 1 のオブジェクト管理方法が、コンポーネント 2 2 に合わせて切り替えられる。

【 0 0 2 4 】

一般には、アプリケーション 1 1 には 1 つ以上のコンポーネントが含まれるので、選択部 3 6 は、コンポーネント毎に適切なオブジェクト管理方法を選択する。そして、切替部 3 7 は、選択されたオブジェクト管理方法に対応するオブジェクトキャッシュ部品とオブジェクト永続化処理部品を配置して、コンポーネント毎のキャッシング機構を生成する。

【 0 0 2 5 】

例えば、アプリケーション 1 1 が 2 つのコンポーネントを含む場合は、図 3 のようなシステムが構築される。図 3 のアプリケーション 1 1 は、コンポーネント 2 2、2 5 を含み、コンポーネント 2 5 は、インタフェース 2 6 および永続化オブジェクト 2 7 を含む。

【 0 0 2 6 】

この場合、選択部 3 6 は、コンポーネント 2 2、2 5 のそれぞれに適したオブジェクト管理方法を選択する。そして、切替部 3 7 は、コンポーネント 2 2、2 5 に対して、それぞれ、オブジェクトキャッシュ部品 3 4、3 9 とオブジェクト永続化処理部品 3 5、4 0 をコンポーネント基盤 1 2 に組み込む。また、インタフェース 3 8 は、インタフェース 3 3 と同様に、インタフェース 2 6 とオブジェクトキャッシュ部品 3 9 の間のインタフェースとして動作する。

【 0 0 2 7 】

このように、アプリケーションの構成単位であるコンポーネント毎にオブジェクトキャッシュ部品とオブジェクト永続化処理部品を切り替えることで、コンポーネント毎に適切なキャッシング機構が生成される。これにより、アプリケーション

ョンの性質に適したオブジェクト管理が実現される。

【 0 0 2 8 】

ただし、切替部 3 7 は、オブジェクトキャッシュ部品とオブジェクト永続化処理部品の両方をコンポーネント毎に切り替える必要はなく、必要に応じて、いずれか一方のみを切り替えることもできる。

【 0 0 2 9 】

ところで、選択部 3 6 および切替部 3 7 がコンポーネント毎のキャッシング機構を生成する方法としては、オブジェクトキャッシュ部品とオブジェクト永続化処理部品を自動生成する第 1 の方法と、あらかじめ用意された部品群の中からオブジェクトキャッシュ部品とオブジェクト永続化処理部品を選択する第 2 の方法が考えられる。第 1 の方法は、主に、アプリケーション 1 1 の構築時に用いられ、第 2 の方法は、主に、アプリケーション 1 1 の運用時に用いられる。

【 0 0 3 0 】

図 4 は、第 1 の方法に基づくシステムを示している。図 4 の切替部 3 7 は、あらかじめオブジェクトキャッシュ部品 3 4 とオブジェクト永続化処理部品 3 5 のプログラムのテンプレートを保持している。選択部 3 6 は、選択したオブジェクト管理方法を切替部 3 7 に通知し、切替部 3 7 は、通知されたオブジェクト管理方法を実現するオブジェクトキャッシュ部品 3 4 とオブジェクト永続化処理部品 3 5 を自動生成して、コンポーネント基盤 1 2 に組み込む。

【 0 0 3 1 】

図 5 は、図 4 のオブジェクト管理部 3 1 による組み込み処理のフローチャートである。ここでは、コンポーネント基盤 1 2 上に処理対象の複数のコンポーネントが存在するものとする。

【 0 0 3 2 】

まず、選択部 3 6 は、コンポーネント基盤 1 2 上の 1 つのコンポーネントについてオブジェクト管理方法を選択する（ステップ S 1）。次に、切替部 3 7 は、テンプレートを用いて、選択されたオブジェクト管理方法に合ったオブジェクトキャッシュ部品のプログラムを生成し（ステップ S 2）、その管理方法に合ったオブジェクト永続化処理部品のプログラムを生成する（ステップ S 3）。そして

、生成した部品をコンポーネント基盤に組み込む（ステップ S 4）。

【 0 0 3 3 】

次に、選択部 3 6 は、コンポーネント基盤 1 2 上に未処理のコンポーネントがあるか否かをチェックし（ステップ S 5）、そのようなコンポーネントがあれば、ステップ S 1 以降の処理を繰り返す。そして、すべてのコンポーネントについてオブジェクト管理方法が選択されると、処理を終了する。

【 0 0 3 4 】

このように、オブジェクトキャッシュ部品とオブジェクト永続化処理部品を自動生成することにより、これらの部品をあらかじめ用意する必要がなくなる。オブジェクトキャッシュ部品とオブジェクト永続化処理部品のプログラムおよびテンプレートの例については後述する。

【 0 0 3 5 】

図 6 は、第 2 の方法に基づくシステムを示している。図 6 のオブジェクト管理部 3 1 は格納部 4 1 を含み、選択部 3 6 は、選択したオブジェクト管理方法を指定する情報を格納部 4 1 に格納する。これにより、コンポーネント 2 2 に適したオブジェクト管理方法が格納部 4 1 に設定される。

【 0 0 3 6 】

切替部 3 7 は、格納部 4 1 からオブジェクト管理方法の情報を読み、あらかじめ用意されたオブジェクトキャッシュ部品群 4 2 およびオブジェクト永続化処理部品群 4 3 から、それぞれ、設定されたオブジェクト管理方法を実現する部品を選択して、コンポーネント基盤 1 2 に組み込む。

【 0 0 3 7 】

図 7 は、図 6 のオブジェクト管理部 3 1 による組み込み処理のフローチャートである。ここでは、コンポーネント基盤 1 2 上に処理対象の複数のコンポーネントが存在するものとする。図 7 のステップ S 1 1、S 1 4、および S 1 5 の処理については、図 5 のステップ S 1、S 4、および S 5 の処理と同様である。

【 0 0 3 8 】

ステップ S 1 1 において、選択部 3 6 がオブジェクト管理方法を選択すると、次に、切替部 3 7 は、選択されたオブジェクト管理方法に合ったオブジェクトキ

キャッシュ部品を部品群 4 2 の中から選択する（ステップ S 1 2）。そして、その管理方法に合ったオブジェクト永続化処理部品を部品群 4 3 の中から選択して（ステップ S 1 3）、ステップ S 1 4 以降の処理を行う。

【 0 0 3 9 】

このように、オブジェクトキャッシュ部品とオブジェクト永続化処理部品を部品群から選択することにより、プログラム自動生成のための処理を組み込む必要がなくなる。あらかじめ用意されるオブジェクトキャッシュ部品とオブジェクト永続化処理部品のプログラムの例については後述する。

【 0 0 4 0 】

選択部 3 6 は、ユーザが指定したオブジェクト管理方法を選択する以外に、アプリケーション 1 1 が指定したオブジェクト管理方法を選択することもできる。図 8 は、このようなシステムを示している。図 8 において、クライアントアプリケーション 2 1 は、コンポーネント 2 2 に適したオブジェクト管理方法を指定し、選択部 3 6 は、その指定を受け付けて、指定されたオブジェクト管理方法を選択する。この選択方法によれば、ユーザは、コンポーネント毎にオブジェクト管理方法を指定する必要がなくなる。

【 0 0 4 1 】

また、選択部 3 6 は、アプリケーションの用途をユーザに入力させ、入力された情報を基に適切なオブジェクト管理方法を選択することもできる。図 9 は、このようなシステムを示している。図 9 の選択部 3 6 は、入力部 5 1、推論部 5 2、および推論ルール格納部 5 3 を含む。

【 0 0 4 2 】

推論ルール格納部 5 3 は、適切なオブジェクト管理方法を推論するための推論ルールを保持している。入力部 5 1 は、ユーザが指定したコンポーネント 2 2 の用途を入力し、推論部 5 2 は、推論ルール格納部 5 3 の推論ルールに従って、入力された用途からコンポーネント 2 2 に適したオブジェクト管理方法を決定する。この選択方法によれば、ユーザは、コンポーネントの用途を指定するだけでよく、オブジェクト管理方法を指定する必要がなくなる。推論ルール格納部 5 3 には、例えば、以下のような推論ルールが格納される。

```

i f   用途=マスタ検索   t h e n
      オブジェクトキャッシュ部品=参照のみ部品
      オブジェクト永続化処理部品=利用データ項目絞込み部品
e l s e   i f   用途=オンライン伝票処理   t h e n
      オブジェクトキャッシュ部品=トランザクション内キャッシュ部品
      オブジェクト永続化処理部品=全データ項目処理部品
e l s e   i f   用途=バッチ伝票処理   t h e n
      オブジェクトキャッシュ部品=シーケンシャル処理部品
      オブジェクト永続化処理部品=利用データ項目絞込み部品
. . .

```

ここで、マスタ検索は、二次記憶 1 3 から元データを読み出す処理に対応し、オンライン伝票処理およびバッチ伝票処理は、業務システムにおける伝票データの参照・更新等の処理に対応する。

【 0 0 4 3 】

また、参照のみ部品は、オブジェクトのデータを更新せずに、参照のみを行うアプリケーションのために用意されたオブジェクトキャッシュ部品を表し、トランザクション内キャッシュ部品は、オブジェクトのデータの更新を行うアプリケーションのために用意されたオブジェクトキャッシュ部品を表す。シーケンシャル処理部品は、検索結果として返された複数のオブジェクトを順番に処理するアプリケーションのために用意されたオブジェクトキャッシュ部品を表す。

【 0 0 4 4 】

また、利用データ項目絞込み部品は、オブジェクトのデータのうち指定された項目についてのみ、二次記憶 1 3 に対する書込み・読出しを行うオブジェクト永続化処理部品を表し、全データ項目処理部品は、オブジェクトのすべてのデータ項目について、二次記憶 1 3 に対する書込み・読出しを行うオブジェクト永続化処理部品を表す。これらの部品の例については、後述することにする。

【 0 0 4 5 】

切替部 37 は、選択部 36 が選択したオブジェクト管理方法に基づいて切替を行う代わりに、あらかじめ決められたスケジュールに従って自動的に切替を行うこともできる。図 10 は、このようなシステムを示している。図 10 のオブジェクト管理部 31 は、選択部 36 に替えて、タイマ 61、切替スケジュール定義部 62、および切替スケジュール格納部 63 を含む。

【0046】

切替スケジュール定義部 62 は、どのオブジェクト管理方法をいつ採用するか（どのオブジェクトキャッシュ部品とオブジェクト永続化処理部品をいつ使うか）を定義する切替スケジュール情報を生成し、切替スケジュール格納部 63 に格納する。また、タイマ 61 は、日時情報を生成して、切替部 37 に出力する。切替部 37 は、タイマ 61 からの日時情報を参照しながら、切替スケジュール格納部 63 の切替スケジュール情報に従って、オブジェクトキャッシュ部品とオブジェクト永続化処理部品を切り替える。

【0047】

例えば、昼間はオンライン伝票処理を行い、夜間はバッチ伝票処理を行うようなシステムの場合、あらかじめ昼間と夜間で異なる部品を使用する切替スケジュールを用意しておく。このように、切替スケジュール情報に従って切替を行うことで、アプリケーションの運用時にオブジェクト管理方法を選択する必要がなくなる。

【0048】

次に、図 11 から図 18 までを参照しながら、オブジェクトキャッシュ部品の例について説明する。

図 11 は、トランザクション内キャッシュ部品の例を示している。図 11 のオブジェクトキャッシュ部品 71 は、オンライン伝票処理のように、オブジェクトのデータを更新するコンポーネントのために用意され、トランザクション毎にオブジェクトをキャッシュするキャッシュテーブルを持つ。そして、トランザクション開始後からキャッシュテーブルにオブジェクトを登録し、トランザクション終了時にキャッシュテーブルをクリアする。

【0049】

ここでは、オブジェクトキャッシュ部品71は、トランザクションA、B、およびCにそれぞれ対応するキャッシュテーブル72、73、および74を保持している。そして、各キャッシュテーブルには識別情報とともに永続化オブジェクトが登録される。このうち、トランザクションC用のキャッシュテーブル74には、識別情報ID1、ID2、およびID3にそれぞれ対応する永続化オブジェクト75、76、および77が登録されている。

【0050】

その後、トランザクションCがコミットまたはロールバックにより終了すると、オブジェクトキャッシュ部品71は、図12に示すように、キャッシュテーブル74をクリアする。

【0051】

このようなオブジェクトキャッシュ部品によれば、永続化オブジェクトがトランザクション毎にキャッシュされ、トランザクション終了時にはキャッシュが解放されるため、常に最新のデータに基づいて処理が行われる。

【0052】

図13は、参照のみ部品の例を示している。図13のオブジェクトキャッシュ部品82は、マスタ検索のように、オブジェクトのデータの参照のみを行うコンポーネントのために用意され、全トランザクション共通のキャッシュテーブルを持つ。そして、1つのトランザクションが終了してもキャッシュテーブルをクリアしない。したがって、他のトランザクションも同じキャッシュテーブルに登録されたオブジェクトを利用することができる。

【0053】

ここでは、オブジェクトキャッシュ部品81が保持するキャッシュテーブル82には、図11のキャッシュテーブル74と同様に、永続化オブジェクト75、76、および77が登録されている。その後、トランザクションCが終了してもキャッシュテーブル82はクリアされず、トランザクションAおよびBは、これらの永続化オブジェクトを参照することができる。

【0054】

このようなオブジェクトキャッシュ部品によれば、永続化オブジェクトが複数

のトランザクションに共通のキャッシュに保持されるので、それらのトランザクションが同じキャッシュを利用することができる。

【0055】

図14は、シーケンシャル処理部品の例を示している。図14のオブジェクトキャッシュ部品91は、バッチ伝票処理のように、オブジェクトの検索結果としてオブジェクトの集合が返されたとき、それらのオブジェクトを必ず順番に処理するようなコンポーネントのために用意される。そして、アプリケーション11が集合の次の要素に対応するオブジェクトを取得しようとしたとき、直前まで使用していたオブジェクトのデータを二次記憶13に書き込み、次の要素のデータをそのオブジェクトに設定して、それをアプリケーション11に返す。

【0056】

まず、アプリケーション11から検索要求があると、オブジェクトキャッシュ部品91は、二次記憶13からデータ集合を取得する。そして、対応するオブジェクトの集合を表す集合オブジェクト92を生成して、それをアプリケーション11に返す。このとき、図2の二次記憶インタフェース32は、二次記憶13からデータ集合の先頭部分または全部（データD1、D2、およびD3）を取得して、一時的に保持する。

【0057】

次に、アプリケーション11は、図15に示すように、集合オブジェクト92の最初のオブジェクトの取得を要求する。このとき、オブジェクトキャッシュ部品91は、二次記憶インタフェース32から最初のデータD1を永続化オブジェクト93に設定し、永続化オブジェクト93をアプリケーション11に返す。

【0058】

次に、アプリケーション11は、図16に示すように、集合オブジェクト92の次のオブジェクトの取得を要求する。このとき、オブジェクトキャッシュ部品91は、永続化オブジェクト93のデータを二次記憶13に書き込む。そして、図17に示すように、二次記憶インタフェース32から2番目のデータD2を永続化オブジェクト93に設定し、永続化オブジェクト93をアプリケーション11に返す。

【 0 0 5 9 】

以下、同様の処理が繰り返され、二次記憶インタフェース 3 2 のデータがすべて永続化オブジェクト 9 3 に設定されると、次の部分データが二次記憶 1 3 から取得される。そして、データ集合のデータがすべて取得されると、処理が終了する。

【 0 0 6 0 】

このようなオブジェクトキャッシュ部品によれば、1 つの永続化オブジェクトを繰り返し利用することで、データ集合をアプリケーションに渡すことができる。したがって、必要なオブジェクトの個数が最小化され、コンポーネント基盤 1 2 の資源の利用効率が向上する。

【 0 0 6 1 】

図 1 8 は、二次記憶 1 3 のすべてのデータをキャッシュするオブジェクトキャッシュ部品を示している。図 1 8 のオブジェクトキャッシュ部品 1 0 1 は、オブジェクトのデータの参照のみを行うコンポーネントのために用意され、参照されるオブジェクトに対応する二次記憶 1 3 のデータ量が少ない場合に用いられる。そして、アプリケーション 1 1 が検索要求を出す前に、あらかじめ二次記憶 1 3 内のすべてのデータをキャッシュに登録しておく。

【 0 0 6 2 】

ここでは、二次記憶 1 3 内のすべてのデータ D 1、D 2、および D 3 がそれぞれ永続化オブジェクト 1 0 3、1 0 4、および 1 0 5 に変換され、キャッシュテーブル 1 0 2 に登録されている。このようなオブジェクトキャッシュ部品によれば、アプリケーション 1 1 からの検索要求があると、直ちにデータを返すことができるため、処理が高速化される。

【 0 0 6 3 】

次に、図 1 9 は、オブジェクト永続化処理部品の 1 つに対応する利用データ項目絞込み部品の例を示している。図 1 9 の利用データ項目格納部 1 1 1 および利用データ項目指示部 1 1 2 は、それぞれ、図 6 の格納部 4 1 および選択部 3 6 に含まれる。利用データ項目指示部 1 1 2 は、コンポーネントの種類毎に利用されるオブジェクトのデータ項目を指定し、その情報を利用データ項目格納部 1 1 1

に格納する。オブジェクトのデータ項目は、例えば、二次記憶 1 3 内のレコードのフィールド（属性）に対応する。

【 0 0 6 4 】

オブジェクトキャッシュ部品 3 4 がキャッシュテーブル 1 1 3 に永続化オブジェクト 1 1 4 を登録するとき、オブジェクト永続化処理部品 1 1 5 は、利用データ項目格納部 1 1 1 を参照し、指定されたデータ項目のみを二次記憶 1 3 から読出して、永続化オブジェクト 1 1 4 に書込む。また、オブジェクトキャッシュ部品 3 4 が永続化オブジェクト 1 1 4 のデータを二次記憶 1 3 に書込むとき、オブジェクト永続化処理部品 1 1 5 は、指定されたデータ項目のみを二次記憶 1 3 に書込む。

【 0 0 6 5 】

ここでは、二次記憶 1 3 内の各データは、A 1、A 2、および A 3 の 3 つの属性から構成されており、オブジェクトキャッシュ部品 3 4 は、識別情報 I D 1 に対応するデータをキャッシュしている。

【 0 0 6 6 】

例えば、利用データ項目格納部 1 1 1 において、これらの 3 つの属性がデータ項目として指定されている場合、オブジェクト永続化処理部品 1 1 5 は、二次記憶 1 3 から I D 1 に対応する 3 つの属性 “ a a a ”、“ x ”、および “ 1 0 0 0 ” を読出し、永続化オブジェクト 1 1 4 の対応する属性に設定する。

【 0 0 6 7 】

また、利用データ項目格納部 1 1 1 において、属性 A 1 のみがデータ項目として指定されている場合、オブジェクト永続化処理部品 1 1 5 は、二次記憶 1 3 から I D 1 の属性 A 1 に対応する “ a a a ” のみを読出し、永続化オブジェクト 1 1 4 の属性 A 1 に設定する。

【 0 0 6 8 】

このようなオブジェクト永続化処理部品によれば、オブジェクトのデータを必要なデータ項目のみに絞込んで、読出し・書込みを行うことができる。このため、二次記憶 1 3 とコンポーネント基盤 1 2 の間のデータ転送が削減され、処理が高速化される。

【 0 0 6 9 】

次に、図 2 0 から図 2 7 までを参照しながら、コンポーネントおよびコンポーネント基盤の具体例について説明する。

図 2 0 は、図 3 のシステムに対応する E J B (Enterprise JavaBeans) (商標) のモデルを示している。ここでは、エンティティビーン (Entity Bean) が永続化オブジェクトに対応し、E J B コンテナ 1 2 1 がコンポーネント基盤 1 2 に対応する。

【 0 0 7 0 】

コンポーネント 2 2 は、E J B O b j e c t 1 2 2 とエンティティビーン 1 2 3 を含み、コンポーネント 2 5 は、E J B O b j e c t 1 2 4 とエンティティビーン 1 2 5 を含む。E J B O b j e c t 1 2 2、1 2 4 は、クライアントアプリケーション 2 1 がエンティティビーン 1 2 3、1 2 5 にアクセスするためのインタフェースである。

【 0 0 7 1 】

また、インタフェース 2 3、3 3 は、E J B H o m e と呼ばれるインタフェースを構成し、クライアントアプリケーション 2 1 からの検索要求等をオブジェクトキャッシュ部品 3 4 に伝える。インタフェース 2 6、3 8 についても同様である。

【 0 0 7 2 】

オブジェクトキャッシュ部品 3 4 は、特定の戦略でエンティティビーンをキャッシュし、オブジェクト永続化処理部品 3 5 は、二次記憶 1 3 内のデータとエンティティビーンの変換を行う。これにより、エンティティビーンのカッシング機構が実現される。

【 0 0 7 3 】

切替部 3 7 が、図 4 のようなプログラムの自動生成を行う場合、例えば、図 2 1 および図 2 2 のようなテンプレートを用いる。図 2 1 および図 2 2 は、それぞれ、オブジェクトキャッシュ部品およびオブジェクト永続化処理部品のテンプレートの例を示している。

【 0 0 7 4 】

図21において、“\$\$Bean\$\$”は、エンティティビーンのクラス名を表し、“\$\$PrimaryKey\$\$”は、エンティティビーンのキーのクラス名を表す。

【0075】

また、図22において、“\$\$BeanFieldColumns\$\$”は、エンティティビーンのデータが格納されるテーブルのカラム名の列を表し、“\$\$Table\$\$”は、エンティティビーンのデータが格納されるテーブルのテーブル名を表す。また、“\$\$PKFieldColumns\$\$”は、エンティティビーンのキーフィールドに対応するカラム名を表す。

【0076】

このテンプレートでは、“//\$\$ foreach \$\$BeanField\$\$”と“//\$\$ end foreach”の間の文が、エンティティビーンのフィールドの数だけ繰り返し展開される。“\$\$Name\$\$”は、エンティティビーンのフィールド名を表し、“\$\$Type\$\$”は、フィールドの型を表し、“\$\$Count\$\$”は、繰り返し回数を表す。

【0077】

図21のテンプレートからは、例えば、図23のようなオブジェクトキャッシュ部品のプログラムが生成される。図23のプログラムでは、“\$\$Bean\$\$”および“\$\$PrimaryKey\$\$”が、それぞれ、“OrderBean”および“OrderBeanPrimaryKey”に置き換えられている。

【0078】

また、図22のテンプレートからは、例えば、図24のようなオブジェクト永続化処理部品のプログラムが生成される。図24のプログラムでは、“\$\$Bean\$\$”が“OrderBean”に置き換えられており、“\$\$BeanFieldColumns\$\$”が“ID, PRODUCT, QUANTITY”に置き換えられている。

【0079】

また、“\$\$Table\$\$”および“\$\$PKFieldColumns\$

\$”は、それぞれ、“ORDERTABLE”および“ID”に置き換えられている。

【0080】

“ORDERTABLE”は、エンティティビーンであるOrderBeanのデータを格納するテーブルのテーブル名であり、“ID”、“PRODUCT”、および“QUANTITY”は、このテーブルのカラム名である。このうち、“ID”がOrderBeanのキーフィールドに対応する。

【0081】

また、“//\$\$ foreach \$\$BeanField\$\$”から“//\$\$ end foreach”までの部分は、OrderBeanのフィールドid、product、およびquantityにデータを設定する文に置き換えられている。

【0082】

また、切替部37が、図6のような部品群を切り替える場合、例えば、図21および図22のような部品があらかじめ用意される。図25および図26は、それぞれ、オブジェクトキャッシュ部品およびオブジェクト永続化処理部品のプログラムの例を示している。

【0083】

図25のオブジェクトキャッシュ部品は、管理されるエンティティビーンの型に依存しない形で実現されている。これに対して、図26のオブジェクト永続化処理部品は、エンティティビーンの情報（フィールドの名前、型等）を表すオブジェクト（ClassDef、FieldDef等）に基づいて、二次記憶13のデータとエンティティビーンのデータの間の変換を行う。

【0084】

ところで、図20のシステムにおいては、オブジェクトキャッシュ部品は、エンティティビーンだけでなく、それに関連するEJBObjectも、選択されたオブジェクト管理方法に合わせて管理することができる。図27は、このようなキャッシング処理を示している。

【0085】

図 2 7 のオブジェクトキャッシュ部品 1 3 1 は、キャッシュテーブル 1 3 2 にエンティティビーンを登録するとき、それに対応する E J B O b j e c t も一緒に登録する。ここでは、I D 1 に対応して、E J B O b j e c t 1 3 3 とエンティティビーン 1 3 4 が登録されており、I D 2 に対応して、E J B O b j e c t 1 3 5 とエンティティビーン 1 3 6 が登録されている。また、I D 3 に対応して、E J B O b j e c t 1 3 7 とエンティティビーン 1 3 8 が登録されている。

【 0 0 8 6 】

このようなキャッシング機構によれば、エンティティビーンだけでなく、E J B O b j e c t も、選択されたオブジェクト管理方法に合わせて再利用することが可能になる。これにより、エンティティビーンとともに、対応する E J B O b j e c t をアプリケーション 1 1 に渡すことが可能になり、アプリケーション 1 1 は、その E J B O b j e c t を介して直ちにエンティティビーンにアクセスすることができる。

【 0 0 8 7 】

なお、本実施形態のオブジェクト管理方法は、上述した E J B に限らず、C O R B A (common object request broker architecture) 等の任意のシステムのコンポーネント基盤に適用することができる。

【 0 0 8 8 】

また、本実施形態のシステムに含まれるクライアントおよびサーバは、例えば、図 2 8 に示すような情報処理装置（コンピュータ）を用いて構成することができる。図 2 8 の情報処理装置は、C P U （中央処理装置）1 4 1、メモリ 1 4 2、入力装置 1 4 3、出力装置 1 4 4、外部記憶装置 1 4 5、媒体駆動装置 1 4 6、およびネットワーク接続装置 1 4 7 を備え、それらはバス 1 4 8 により互いに接続されている。

【 0 0 8 9 】

メモリ 1 4 2 は、例えば、R O M (read only memory)、R A M (random access memory) 等を含み、処理に用いられるプログラムとデータを格納する。C P U 1 4 1 は、メモリ 1 4 2 を利用してプログラムを実行することにより、必要な処理を行う。

【 0 0 9 0 】

この場合、図 2 のアプリケーション 1 1、コンポーネント基盤 1 2、クライアントアプリケーション 2 1、コンポーネント 2 2、インタフェース 2 3、3 3、永続化オブジェクト 2 4、オブジェクト管理部 3 1、二次記憶インタフェース 3 2、オブジェクトキャッシュ部品 3 4、オブジェクト永続化処理部品 3 5、選択部 3 6、および切替部 3 7 と、図 1 0 のタイマ 6 1 および切替スケジュール定義部 6 2 は、プログラムとしてメモリ 1 4 2 に格納される。また、メモリ 1 4 2 は、図 6 の格納部 4 1、図 9 の推論ルール格納部 5 3、図 1 0 の切替スケジュール格納部 6 3、およびキャッシュテーブル等を保持する。

【 0 0 9 1 】

入力装置 1 4 3 は、例えば、キーボード、ポインティングデバイス、タッチパネル等であり、オペレータ（ユーザまたはサーバの管理者）からの指示や情報の入力に用いられる。出力装置 1 4 4 は、例えば、ディスプレイ、プリンタ、スピーカ等であり、オペレータへの問い合わせや処理結果の出力に用いられる。

【 0 0 9 2 】

外部記憶装置 1 4 5 は、例えば、磁気ディスク装置、光ディスク装置、光磁気ディスク（magneto-optical disk）装置、テープ装置等であり、図 2 の二次記憶 1 3 として用いられる。また、情報処理装置は、この外部記憶装置 1 4 5 に、上述のプログラムとデータを保存しておき、必要に応じて、それらをメモリ 1 4 2 にロードして使用する。

【 0 0 9 3 】

媒体駆動装置 1 4 6 は、可搬記録媒体 1 4 9 を駆動し、その記録内容にアクセスする。可搬記録媒体 1 4 9 としては、メモ리카ード、フロッピーディスク、CD-ROM（compact disk read only memory）、光ディスク、光磁気ディスク等、任意のコンピュータ読み取り可能な記録媒体が用いられる。オペレータは、この可搬記録媒体 1 4 9 に上述のプログラムとデータを格納しておき、必要に応じて、それらをメモリ 1 4 2 にロードして使用する。

【 0 0 9 4 】

ネットワーク接続装置 1 4 7 は、LAN（local area network）等の任意の通

信ネットワークに接続され、通信に伴うデータ変換を行う。また、情報処理装置は、上述のプログラムとデータをネットワーク接続装置 1 4 7 を介して他の装置から受け取り、必要に応じて、それらをメモリ 1 4 2 にロードして使用する。

【 0 0 9 5 】

図 2 9 は、図 2 8 の情報処理装置にプログラムとデータを供給することのできるコンピュータ読み取り可能な記録媒体を示している。可搬記録媒体 1 4 9 や外部のデータベース 1 5 0 に保存されたプログラムとデータは、メモリ 1 4 2 にロードされる。そして、CPU 1 4 1 は、そのデータを用いてそのプログラムを実行し、必要な処理を行う。

【 0 0 9 6 】

【発明の効果】

本発明によれば、コンポーネントを組み合わせるアプリケーションを構築し運用するシステムにおいて、個々のアプリケーションに適したキャッシング機構が、コンポーネント基盤により自動的に提供される。これにより、アプリケーション側でキャッシング機構を用意する必要がなくなり、アプリケーション開発が効率化される。

【図面の簡単な説明】

【図 1】

本発明のオブジェクト管理装置の原理図である。

【図 2】

第 1 のシステムの構成図である。

【図 3】

第 2 のシステムの構成図である。

【図 4】

第 3 のシステムの構成図である。

【図 5】

第 1 の組み込み処理のフローチャートである。

【図 6】

第 4 のシステムの構成図である。

【図 7】

第 2 の組み込み処理のフローチャートである。

【図 8】

第 5 のシステムの構成図である。

【図 9】

第 6 のシステムの構成図である。

【図 1 0】

第 7 のシステムの構成図である。

【図 1 1】

トランザクション内キャッシュ部品を示す図である。

【図 1 2】

トランザクション終了時のキャッシュテーブルを示す図である。

【図 1 3】

参照のみ部品を示す図である。

【図 1 4】

シーケンシャル処理部品を示す図である。

【図 1 5】

最初のオブジェクトの取得を示す図である。

【図 1 6】

データの書込みを示す図である。

【図 1 7】

2 番目のオブジェクトの取得を示す図である。

【図 1 8】

全データのキャッシングを示す図である。

【図 1 9】

利用データ項目絞込み部品を示す図である。

【図 2 0】

第 8 のシステムの構成図である。

【図 2 1】

オブジェクトキャッシュ部品のテンプレートの例を示す図である。

【図 2 2】

オブジェクト永続化処理部品のテンプレートの例を示す図である。

【図 2 3】

オブジェクトキャッシュ部品の生成プログラム例を示す図である。

【図 2 4】

オブジェクト永続化処理部品の生成プログラム例を示す図である。

【図 2 5】

オブジェクトキャッシュ部品の例を示す図である。

【図 2 6】

オブジェクト永続化処理部品の例を示す図である。

【図 2 7】

インタフェースのキャッシングを示す図である。

【図 2 8】

情報処理装置の構成図である。

【図 2 9】

記録媒体を示す図である。

【符号の説明】

1 選択手段

2 切替手段

1 1 アプリケーション

1 2 コンポーネント基盤

1 3 二次記憶

2 1 クライアントアプリケーション

2 2、2 5 コンポーネント

2 3、2 6、3 3、3 8 インタフェース

2 4、2 7、7 5、7 6、7 7、9 3、1 0 3、1 0 4、1 0 5、1 1 4 永

続化オブジェクト

3 1 オブジェクト管理部

32 二次記憶インタフェース

34、39、42、71、81、91、101、131 オブジェクトキャッシュ部品

35、40、43、115 オブジェクト永続化処理部品

36 選択部

37 切替部

41 格納部

51 入力部

52 推論部

53 推論ルール格納部

61 タイマ

62 切替スケジュール定義部

63 切替スケジュール格納部

72、73、74、82、102、113、132 キャッシュテーブル

92 集合オブジェクト

111 利用データ項目格納部

112 利用データ項目指示部

121 EJBコンテナ

122、124、133、135、137 EJBObject

123、125、134、136、138 エンティティビーン

141 CPU

142 メモリ

143 入力装置

144 出力装置

145 外部記憶装置

146 媒体駆動装置

147 ネットワーク接続装置

148 バス

149 可搬記録媒体

特 2 0 0 0 - 1 2 9 6 5 5

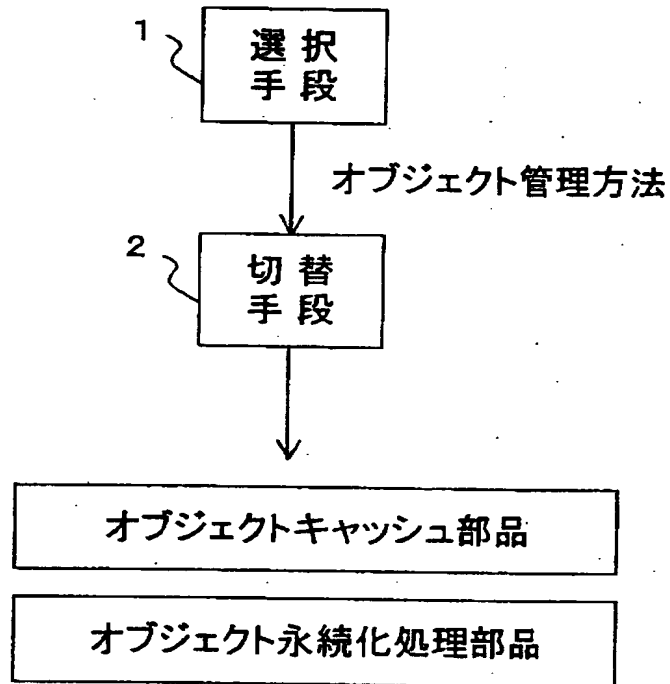
1 5 0 データベース

【書類名】

図面

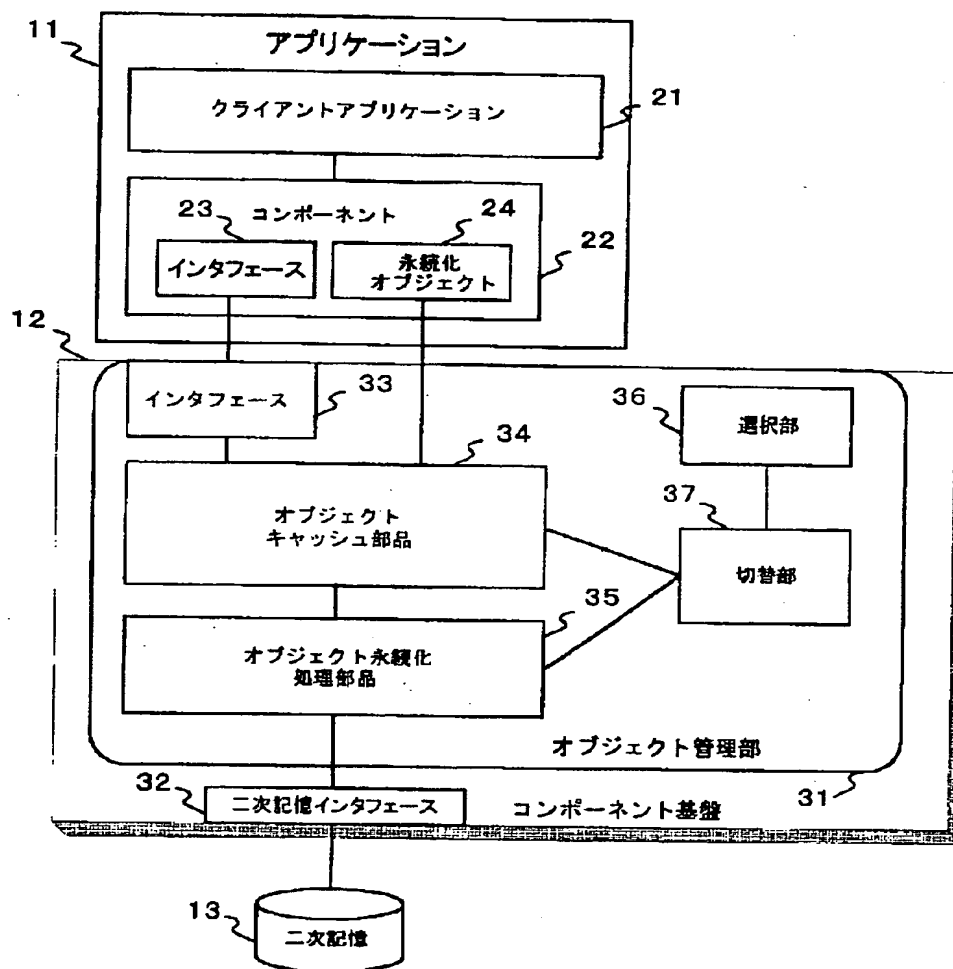
【図 1】

本 発 明 の 原 理 図



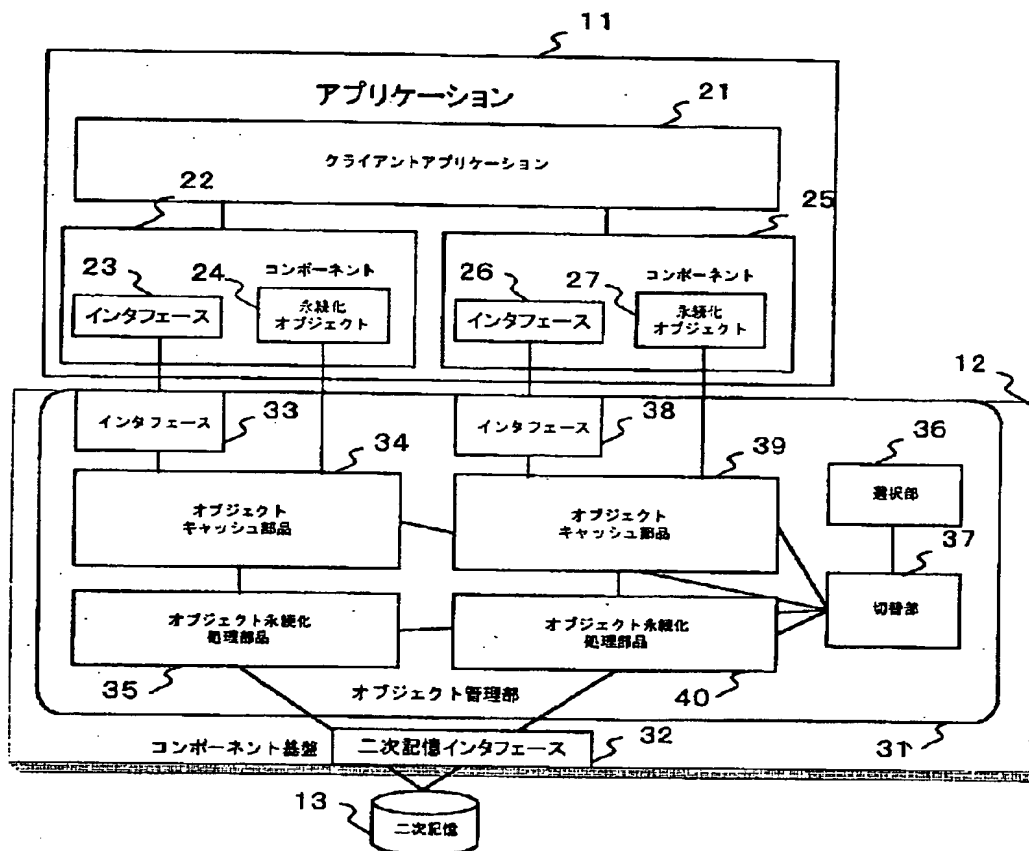
【図2】

第 1 の シ ス テ ム の 構 成 図



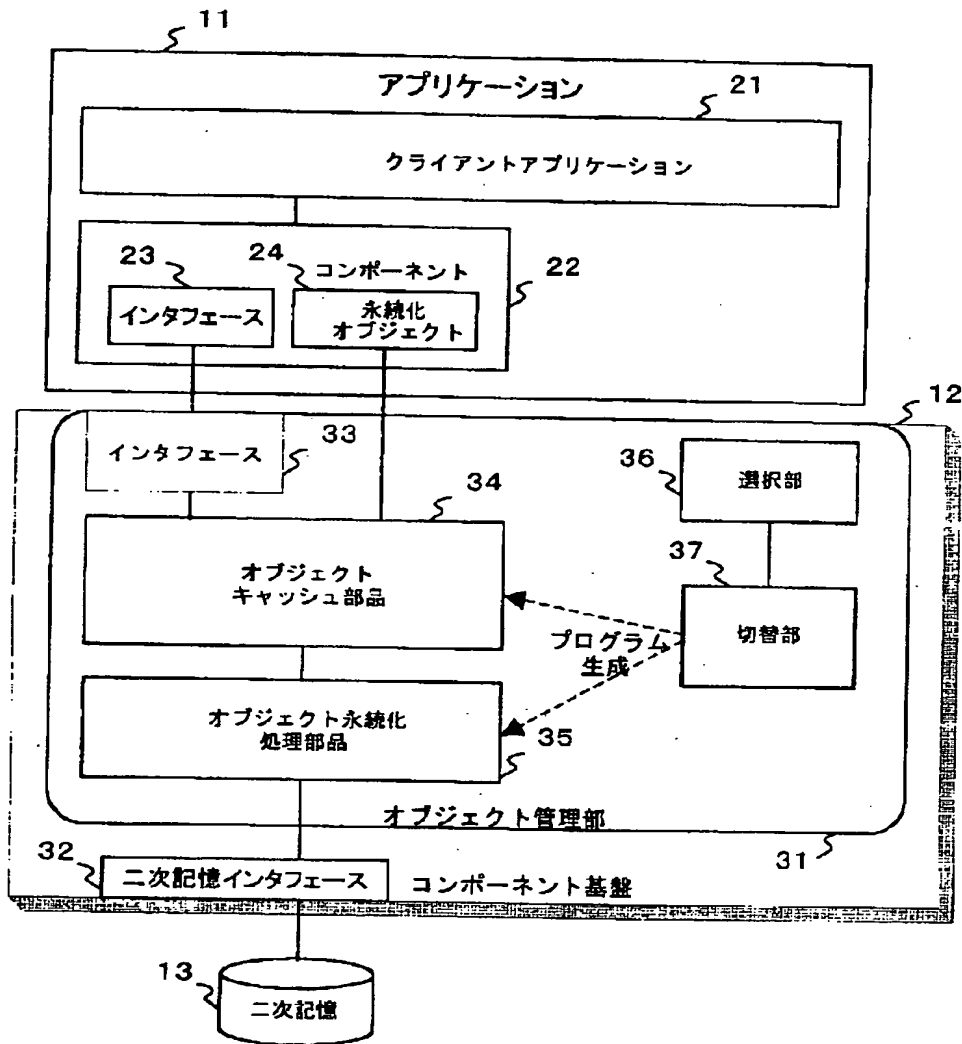
【図3】

第2のシステムの構成図



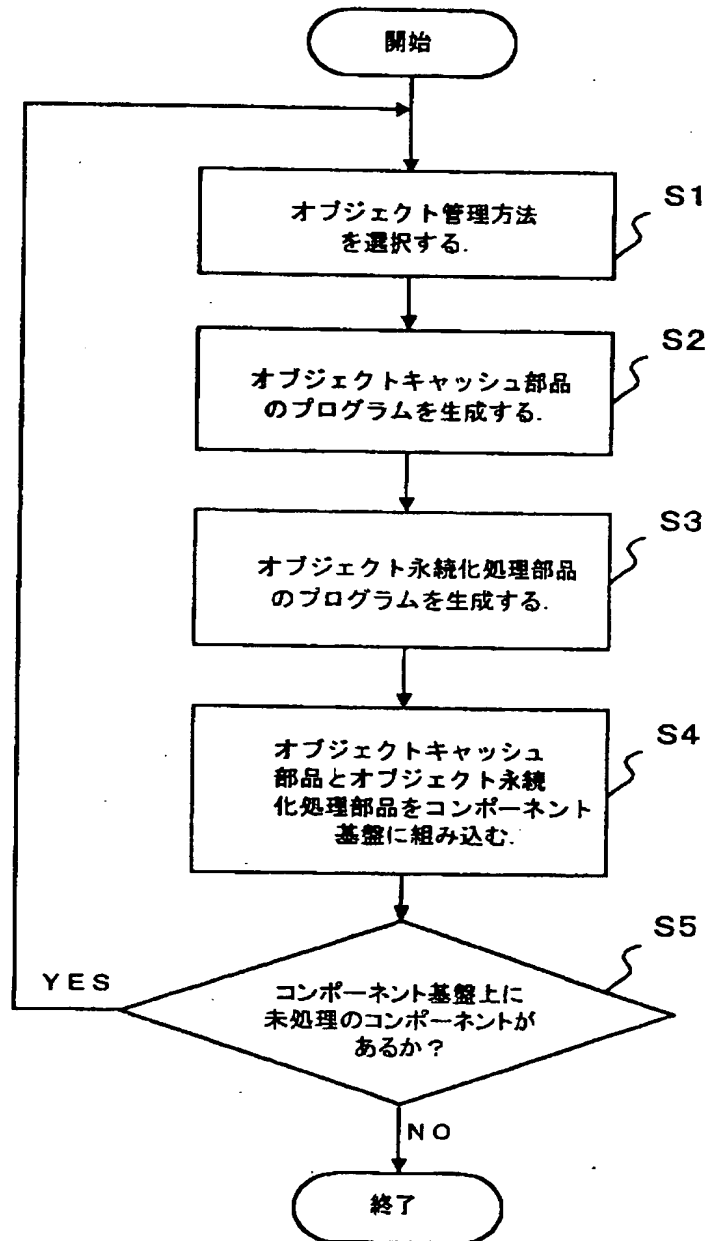
【図4】

第 3 の シ ス テ ム の 構 成 図



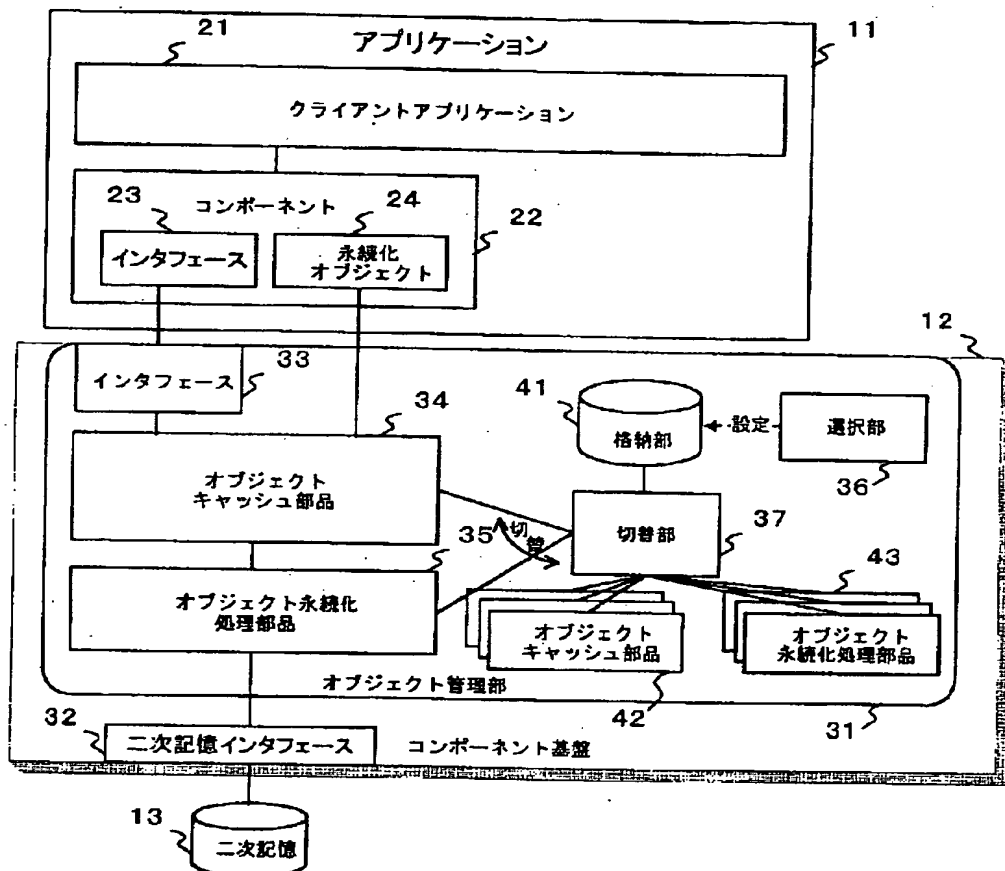
【図5】

第1の組み込み処理のフローチャート



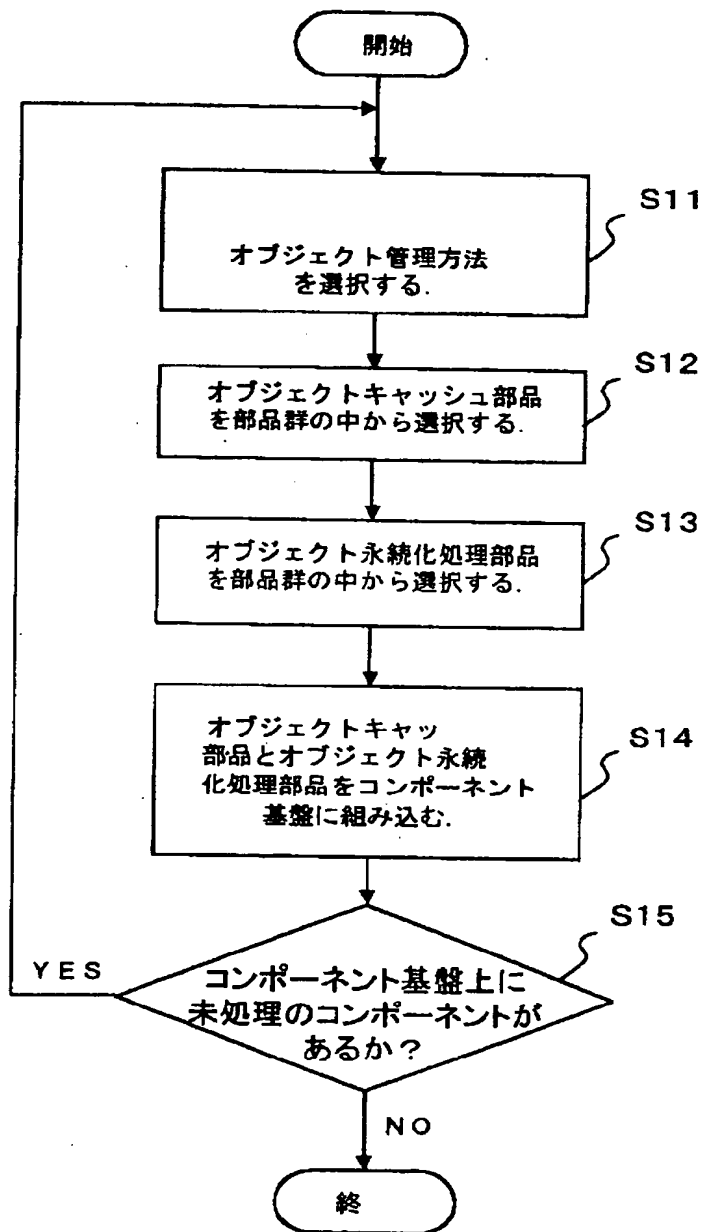
【図6】

第4のシステムの構成図



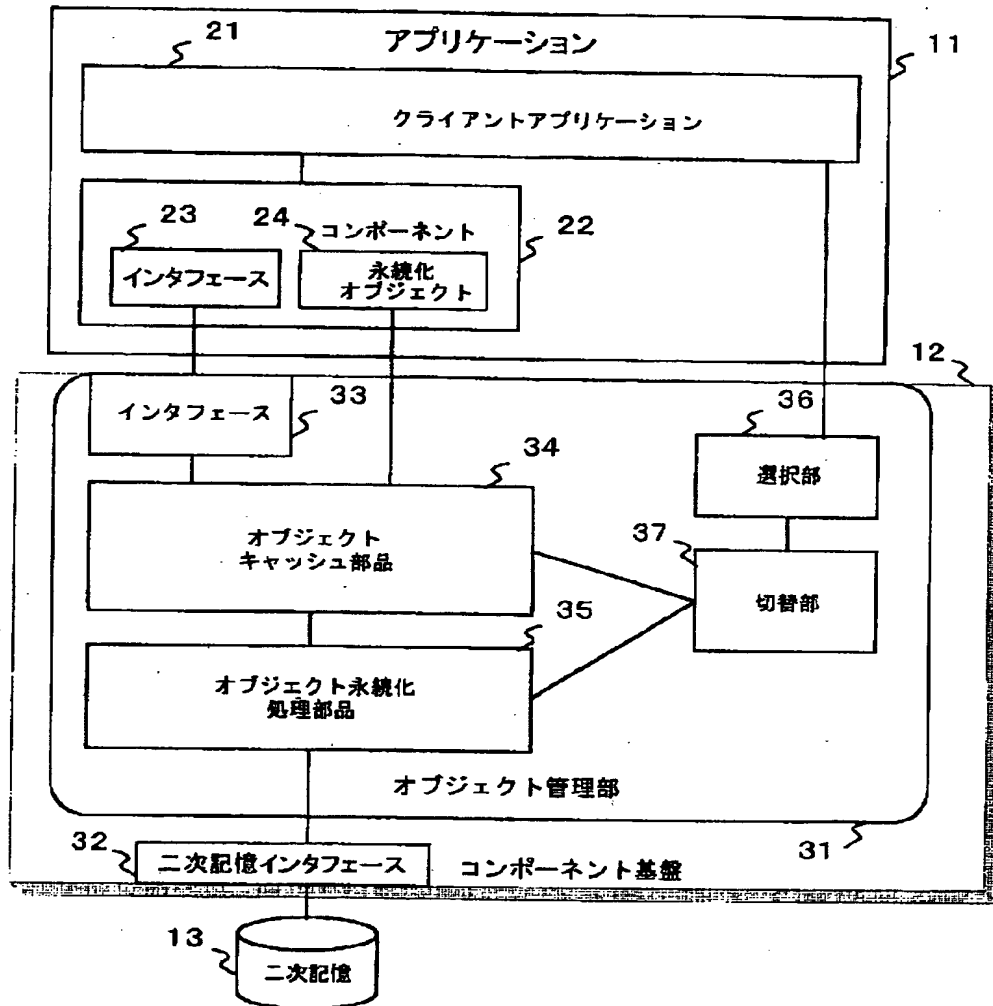
【図 7】

第2の組み込み処理のフローチャート



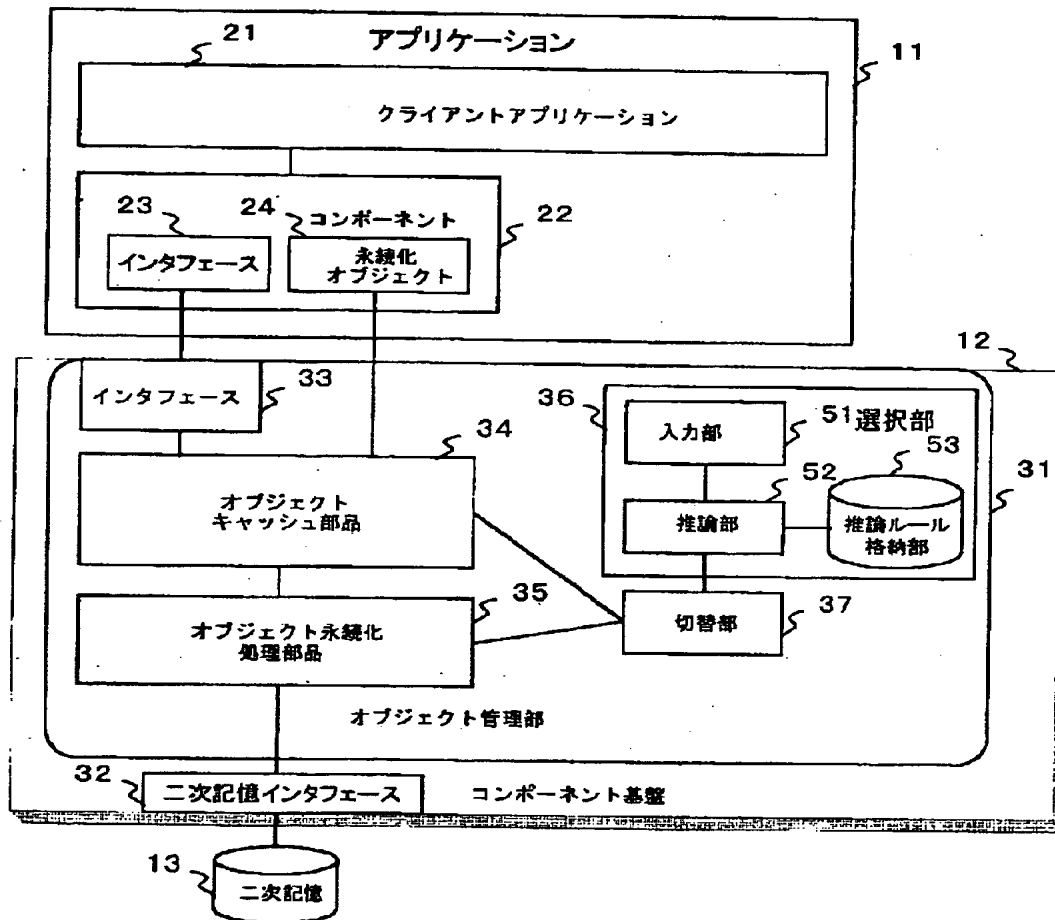
【図 8】

第 5 の シ ス テ ム の 構 成 図



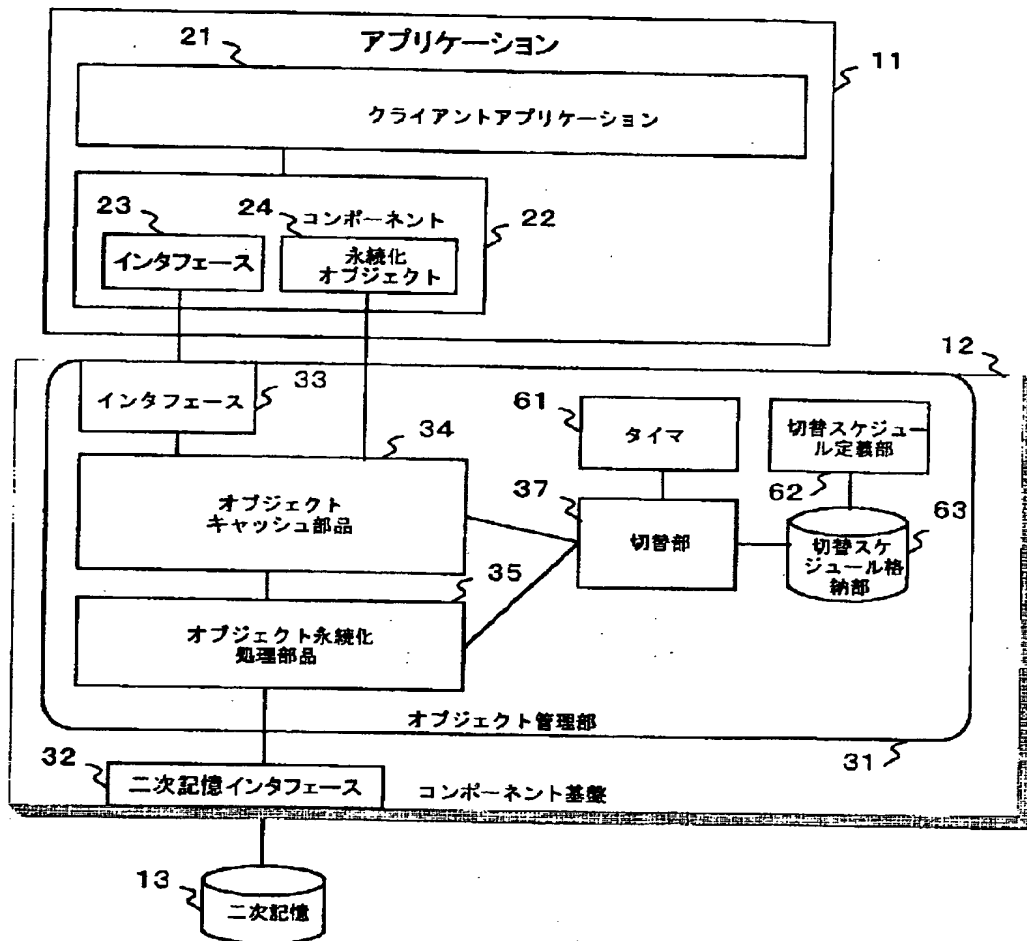
【図9】

第 6 の シ ス テ ム の 構 成 図



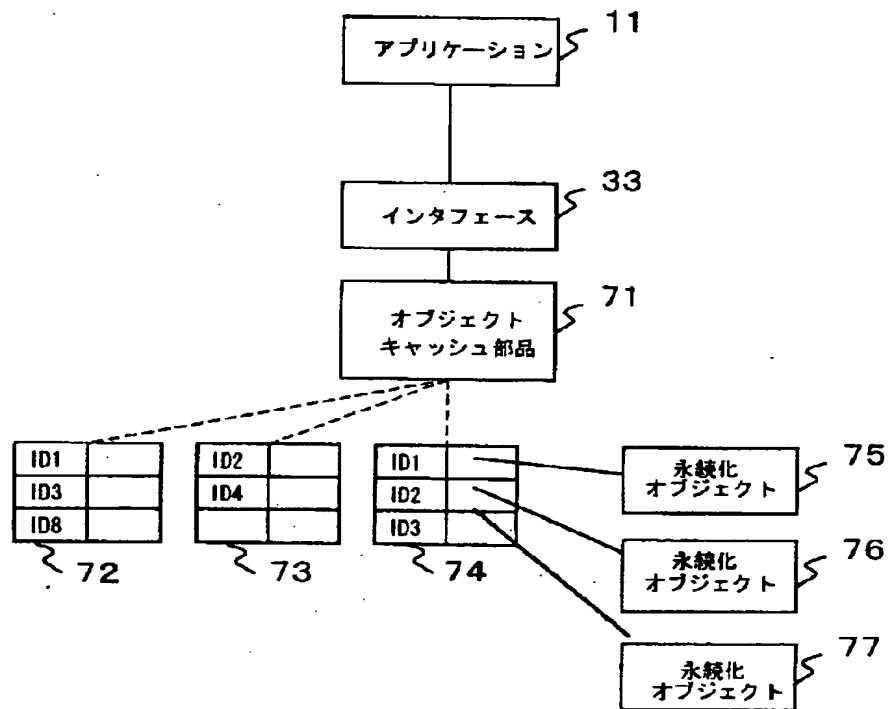
【図10】

第 7 の シ ス テ ム の 構 成 図



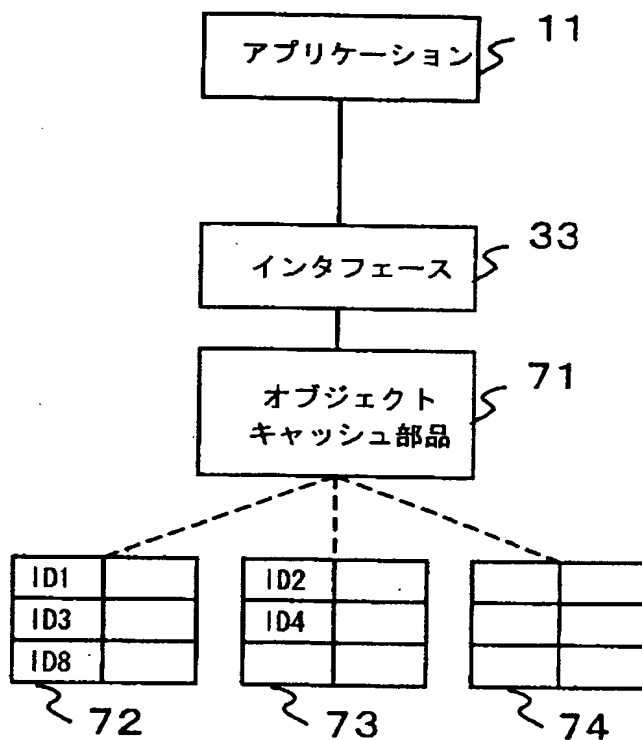
【図 11】

トランザクション内キャッシュ部品を示す図



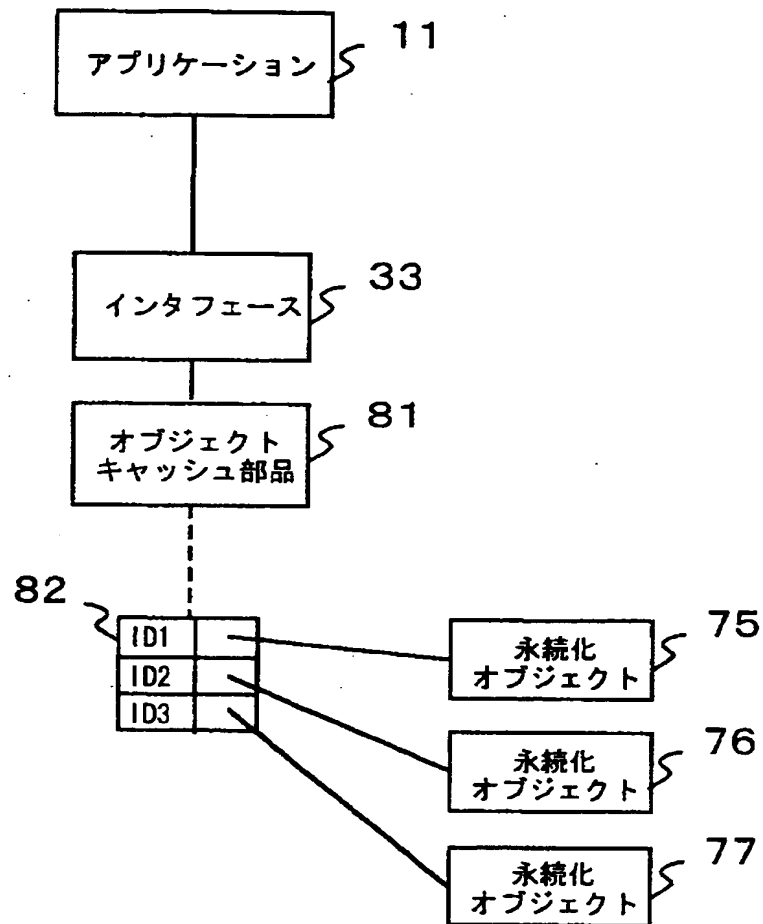
【図 1 2】

トランザクション終了時の
キャッシュテーブルを示す図



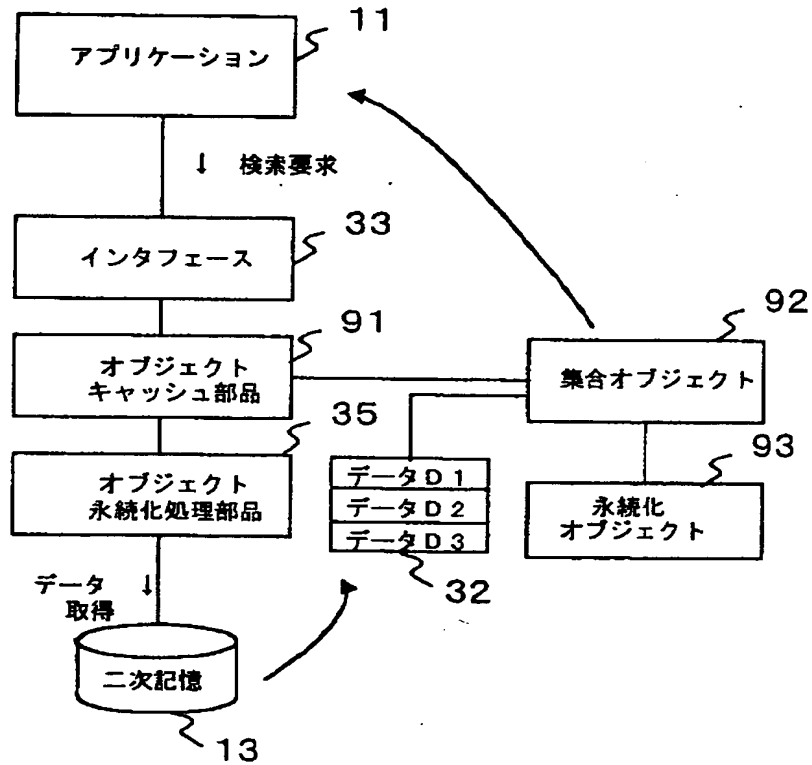
【図13】

参 照 の み 部 品 を 示 す 図



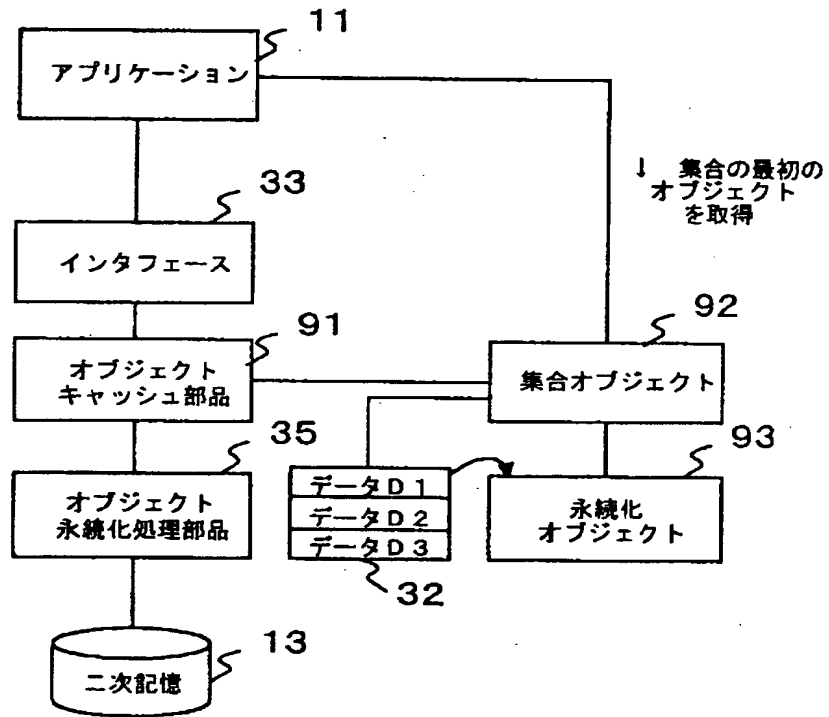
【図 14】

シーケンシャル処理部品を示す図



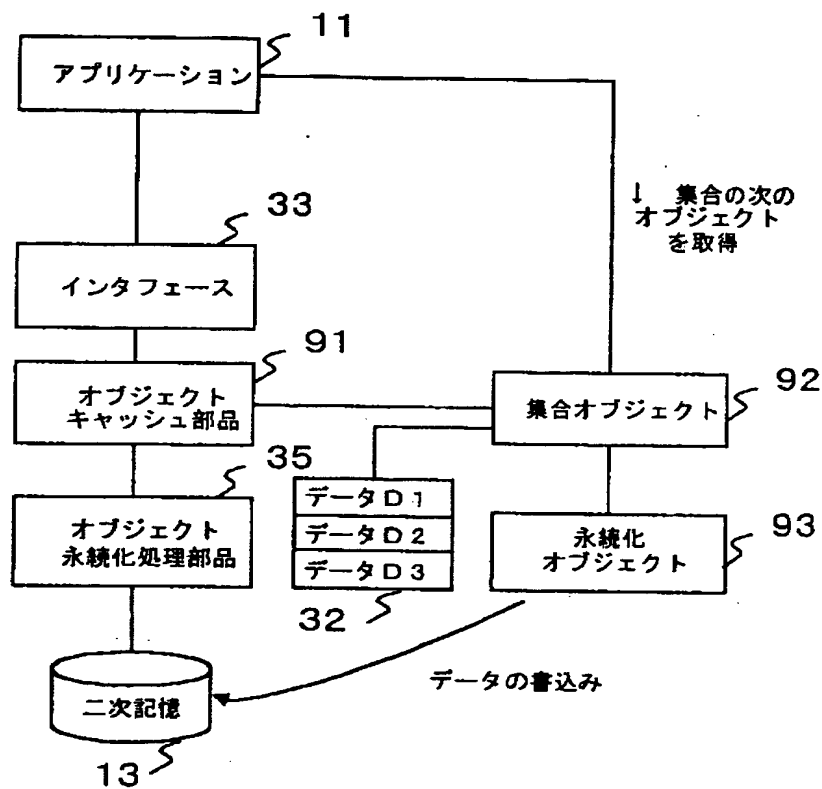
【図 15】

最初のオブジェクトの取得を示す図



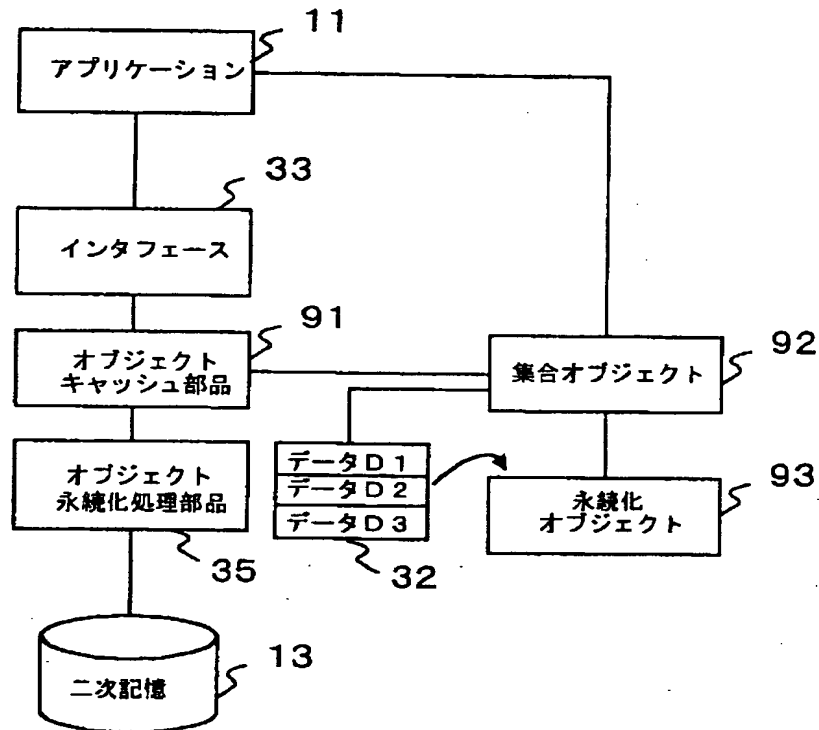
【図16】

データの書込みを示す図



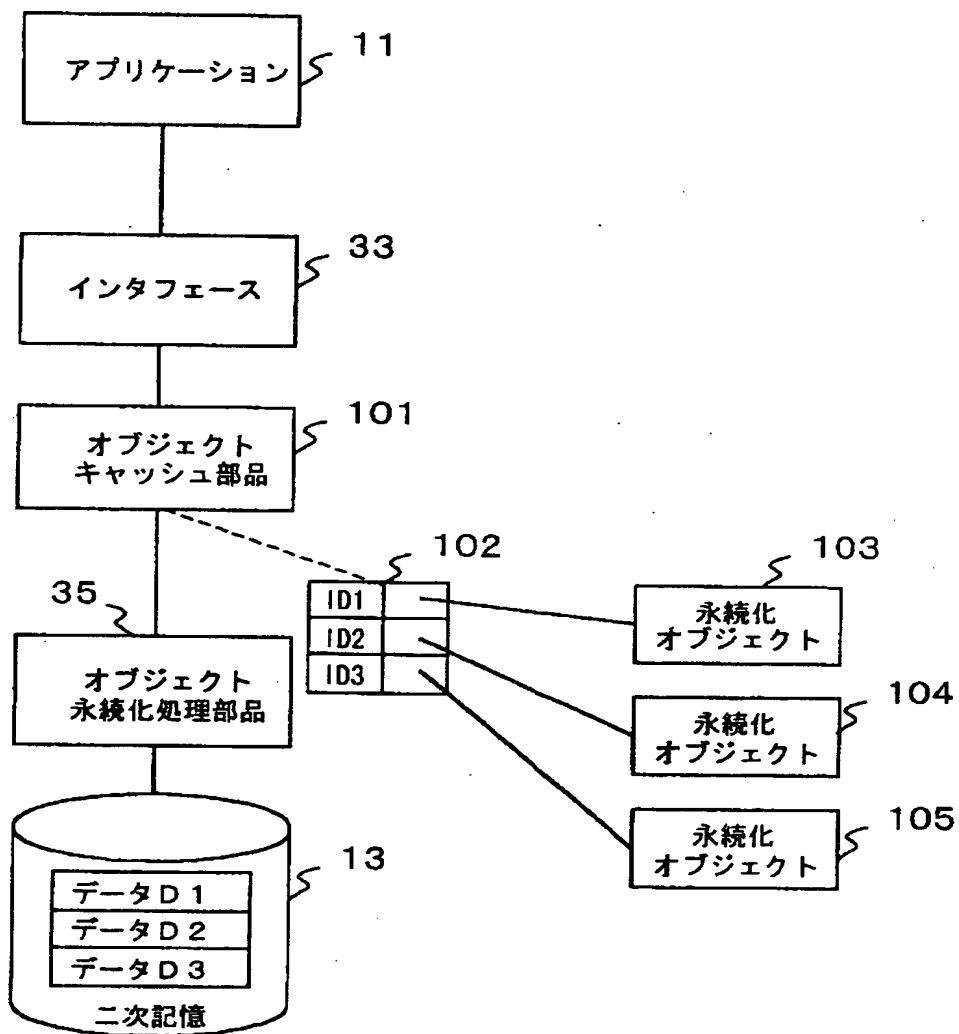
【図 17】

2番目のオブジェクトの取得を示す図



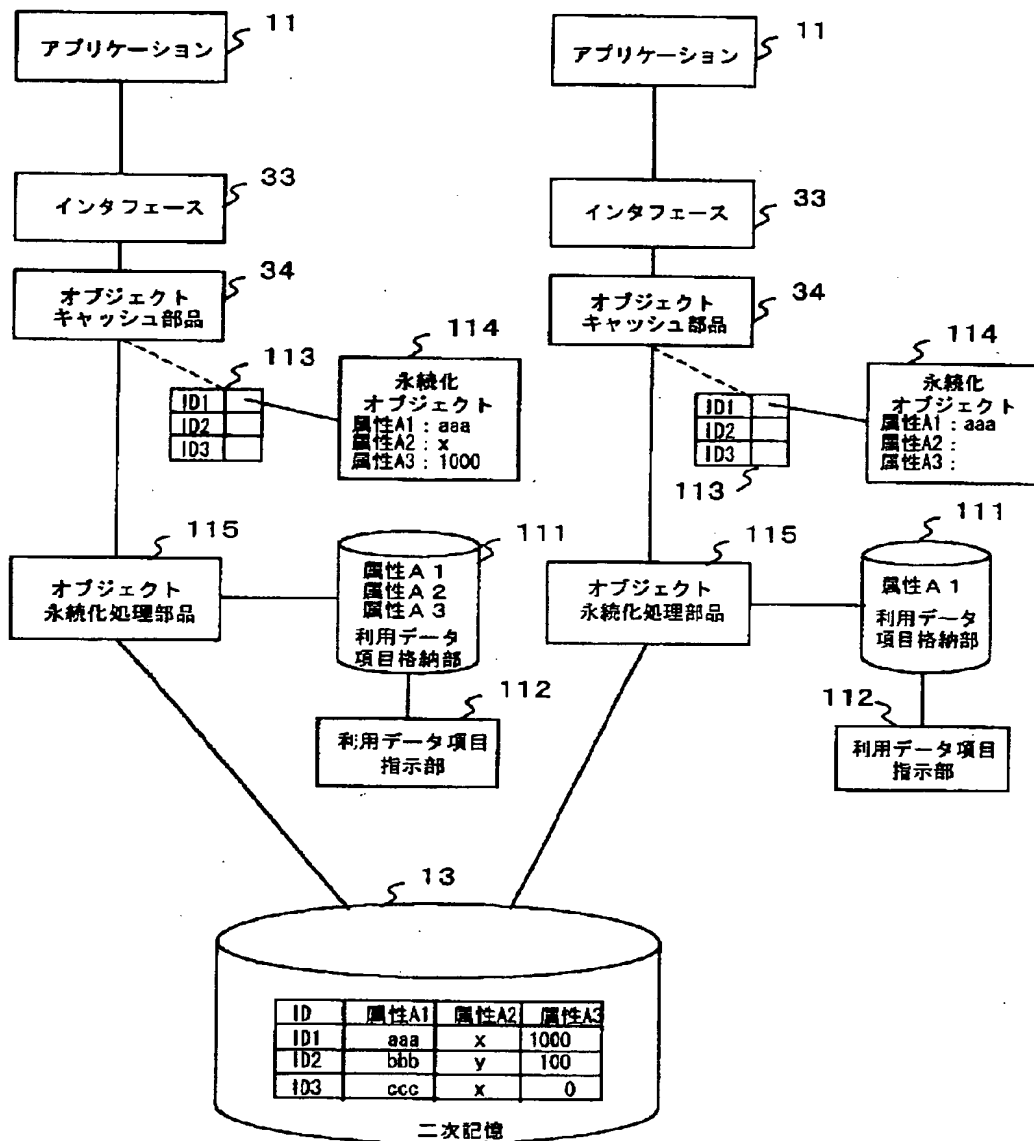
【図 18】

全データのキャッシングを示す図



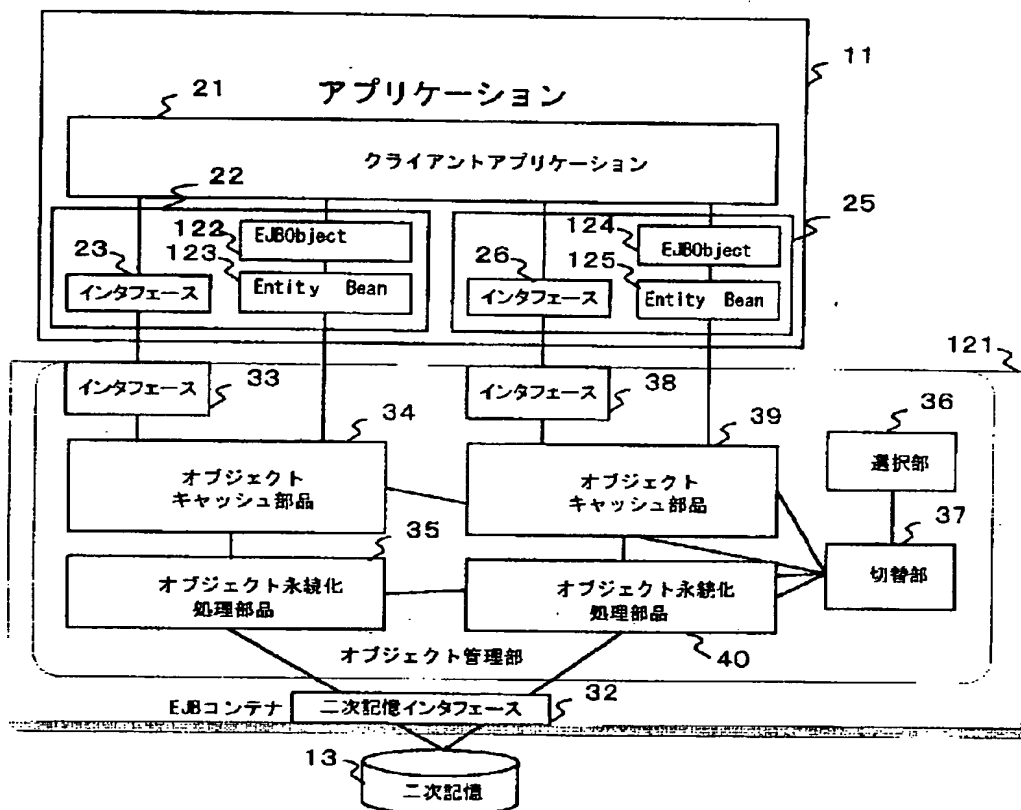
【図19】

利用データ項目絞り込み部品を示す図



【図 20】

第 8 の シ ス テ ム の 構 成 図



【図21】

オブジェクトキャッシュ部品の
テンプレートの例を示す図

```
public class $$Bean$$Cache
{
    Hashtable cacheTableHash;
    $$Bean$$Persistence objPersistence;

    $$Bean$$ findByPrimaryKey($$PrimaryKey$$ pk)
    {
        String transactionID = getTransactionID();
        Hashtable cacheTable = (Hashtable)cacheTableHash.get(transactionID);
        $$Bean$$ bean = cacheTable.get(pk);

        if (bean != null) return bean;

        bean = objPersistence.findBean(pk);
        cacheTable.put(pk, bean);
        return bean;
    }
    ....
}
```

【図 22】

オブジェクト永続化処理部品の
テンプレートの例を示す図

```

public class $$Bean$$Persistence
{
    DataSource dataSource;

    $$Bean$$ findBean($$PrimaryKey$$ pk)
    {
        $$Bean$$ bean = null;
        Connection connection = getConnection();
        String sql = "SELECT $$BeanFieldColumns$$ FROM $$Table$$ WHERE
            $$PKFieldColumns$$ = ?";
        Statement statement = connection.prepareStatement(sql);

        $$ foreach $$PKFields$$
        statement.set$$Type$$($$Count$$,pk.$$Name$$);
        $$ end foreach

        ResultSet rs = statement.executeQuery();
        while(rs.next())
        {
            bean = new $$Bean$$();
            $$ foreach $$BeanFields$$
            bean.$$Name$$ = rs.get$$Type$$($$Count$$);
            $$ end foreach
        }
        return bean;
    }
    ...
}

```

【図 23】

オブジェクトキャッシュ部品の
生成プログラム例を示す図

```
public class OrderBeanCache
{
    Hashtable cacheTableHash;
    OrderBeanPersistence objPersistence;

    OrderBean findByPrimaryKey(OrderBeanPrimaryKey pk)
    {
        String transactionID = getTransactionID();
        Hashtable cacheTable = (Hashtable)cacheTableHash.get(transactionID);
        OrderBean bean = cacheTable.get(pk);

        if (bean != null) return bean;

        bean = objPersistence.findBean(pk);
        cacheTable.put(pk, bean);
        return bean;
    }
    ....
}
```


【図 24】

オブジェクト永続化処理部品の
生成プログラム例を示す図

```
public class OrderBeanPersistence
{
    DataSource dataSource;

    OrderBean findBean(OrderBeanPrimary Key pk)
    {
        OrderBean bean = null;
        Connection connection = getConnection();
        String sql = "SELECT ID,PRODUCT,QUANTITY FROM ORDERTABLE
        WHERE ID = ?";
        Statement statement = connection.prepareStatement(sql);

        statement.setString(1,pk.id);
        statement.setString(2,pk.product);
        statement.setInt(3,pk.quantity);

        ResultSet rs = statement.executeQuery();
        while(rs.next())
        {
            bean = new OrderBean();
            bean.id = rs.getString(1);
            bean.product = rs.getString(2);
            bean.quantity = rs.getInt(3);
        }
        return bean;
    }
    ...
}
```

【図 2 5】

オブジェクトキャッシュ 部品の例を示す図

```
public class ObjectCacheOption1 extends ObjectCache
{
    Hashtable cacheTableHash;
    ObjectPersistence objPersistence;

    Object findByPrimaryKey(Object pk, BeanDef beanDef)
    {
        String transactionID = getTransactionID();
        Hashtable cacheTable = (Hashtable)cacheTableHash.get(transactionID);
        Object bean = cacheTable.get(pk);

        if (bean != null) return bean;

        bean = objPersistence.findBean(pk,beanDef,"findByPrimaryKey");
        cacheTable.put(pk,bean);
        return bean;
    }
    ....
}
```

【図 2 6】

オブジェクト永続化処理部品の例を示す図

```

public class ObjectPersistenceOption1 extends ObjectPersistence
{
    DataSource dataSource;

    Object findBean(Object pk, BeanDef beanDef, String finderName)
    {
        Object bean = null;
        Connection connection = getConnection();
        String sql = beanDef.getSQL(finderName);
        Statement statement = connection.prepareStatement(sql);

        Enumeration fields = beanDef.getFields();
        while (fields.hasMoreElements())
        {
            FieldDef field = (FieldDef)fields.nextElement();
            setValue(statement, field);
        }

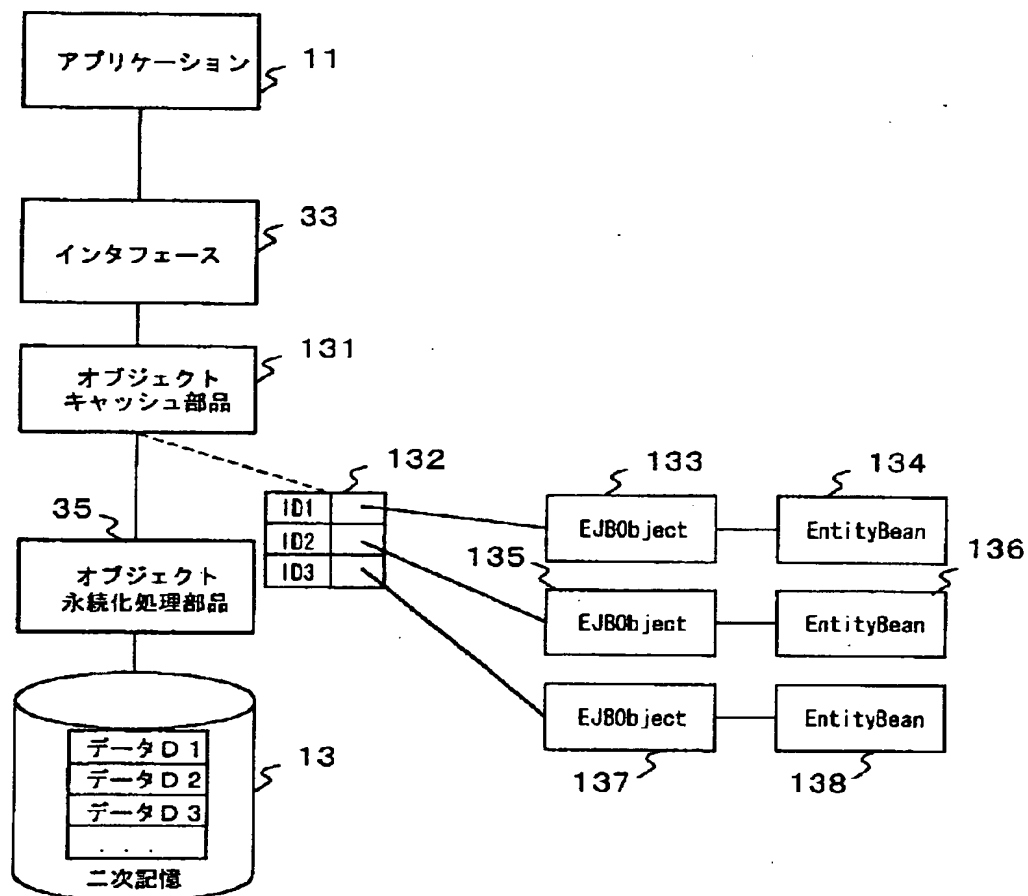
        ResultSet rs = statement.executeQuery();
        while(rs.next())
        {
            bean = beanDef.newInstance();
            fields = beanDef.getFields();
            while (fields.hasMoreElements())
            {
                FieldDef field = (FieldDef)fields.nextElement();
                setValue(bean, rs, field);
            }
        }
        return bean;
    }
    ...

    void setValue(Object bean, ResultSet, FieldDef field)
    {
        Field f = field.getField();
        switch (field.fieldType)
        {
            case FT_INT:
                f.setInt(bean, rs.getInt(field.getColumn()));
                break;
            case FT_LONG:
                f.setLong(bean, rs.getLong(field.getColumn()));
                break;
            ...
        }
    }
    ...
}

```

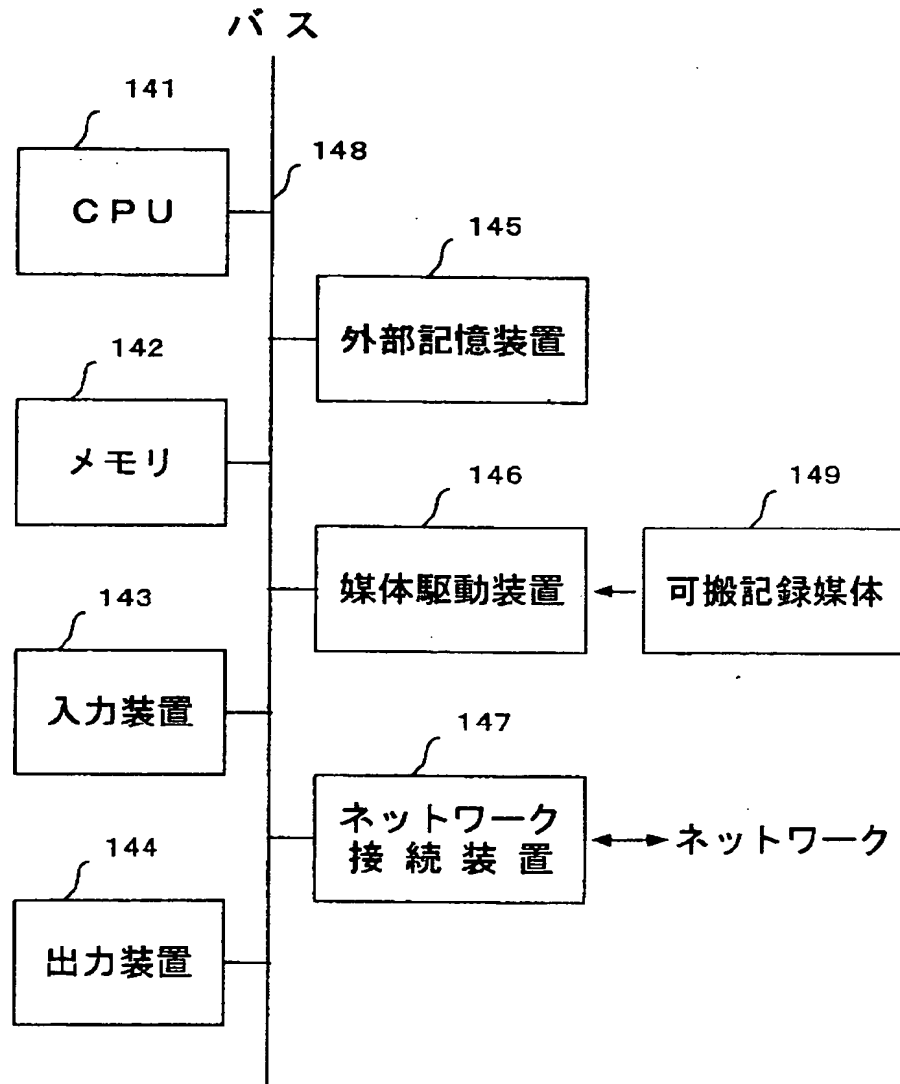
【図 27】

インタフェースのキャッシングを示す図



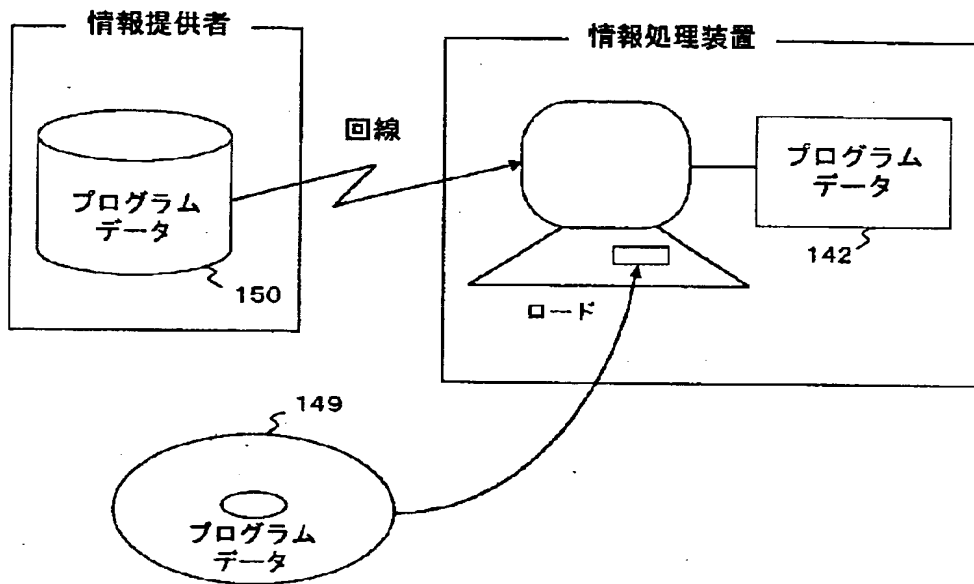
【図 2 8】

情 報 処 理 装 置 の 構 成 図



【図29】

記録媒体を示す図



【書類名】 要約書

【要約】

【課題】 コンポーネントを組み合わせてアプリケーションを構築し運用するシステムにおいて、個々のアプリケーションに適したキャッシング機構をコンポーネント基盤の中で提供することが課題である。

【解決手段】 オブジェクトキャッシュ部品 3 4 は、オブジェクトの種類毎に特定の戦略でオブジェクトをキャッシュし、オブジェクト永続化処理部品 3 5 は、二次記憶 1 3 内のデータとオブジェクトの間の変換を行う。選択部 3 6 は、複数のオブジェクト管理方法の中からコンポーネント 2 2 の種類に適した方法を選択し、切替部 3 7 は、選択された方法に対応するオブジェクトキャッシュ部品 3 4 とオブジェクト永続化処理部品 3 5 をコンポーネント基盤 1 2 に組み込む。

【選択図】 図 2

出 願 人 履 歴 情 報

識別番号 [000005223]

1. 変更年月日 1996年 3月26日

[変更理由] 住所変更

住 所 神奈川県川崎市中原区上小田中4丁目1番1号
氏 名 富士通株式会社